

# Advances in Artificial Life

## ECAL 2013

2-6 September 2013, Taormina, Italy

Proceedings of the twelfth European Conference on the Synthesis and Simulation of Living Systems



edited by

Pietro Liò, Orazio Miglino, Giuseppe Nicosia,  
Stefano Nolfi, Mario Pavone



## Attribution-NonCommercial-NoDerivs 3.0 United States

### You are free:



**to Share** — to copy, distribute and transmit the work

### Under the following conditions:



**Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).



**Noncommercial** — You may not use this work for commercial purposes.



**No Derivative Works** — You may not alter, transform, or build upon this work.

### With the understanding that:

**Waiver** — Any of the above conditions can be **waived** if you get permission from the copyright holder.

**Public Domain** — Where the work or any of its elements is in the **public domain** under applicable law, that status is in no way affected by the license.

**Other Rights** — In no way are any of the following rights affected by the license:

- Your fair dealing or **fair use** rights, or other applicable copyright exceptions and limitations;
- The author's **moral** rights;
- Rights other persons may have either in the work itself or in how the work is used, such as **publicity** or privacy rights.

**Notice** — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to <http://creativecommons.org/licenses/by-nc-nd/3.0/us/>.

This is a human-readable summary of the [Legal Code \(the full license\)](http://creativecommons.org/licenses/by-nc-nd/3.0/us/legalcode) which can be found at <http://creativecommons.org/licenses/by-nc-nd/3.0/us/legalcode>.



# Table of Contents

## **Preface**

xiv

## **ECAL - General Track**

Simon Tudge, Richard Watson and Markus Brede. <i>Cooperation and the Division of Labour</i>	1
Yutetsu Kuruma, Hideaki Matsubayashi and Takuya Ueda. <i>Autonomous construction of synthetic cell membrane</i>	9
Jared Moore, Anne Gutmann, Craig McGowan and Philip McKinley. <i>Exploring the Role of the Tail in Bipedal Hopping through Computational Evolution</i>	11
Kiril Kiryazov and Robert Lowe. <i>The role of arousal in embodying the cueXdeficit model in multi-resource human-robot interaction</i>	19
Simon Alderton and Jason Noble. <i>Simulating Sleeping Sickness: a two host agent-based model</i>	27
Adam Erskine and J. Michael Herrmann. <i>Cell Division Behaviour in a Heterogeneous Swarm Environment</i>	35
Guillaume Beslon, B��r��nice Batut, David P. Parsons, Dominique Schneider and Carole Knibbe. <i>An alife game to teach evolution of antibiotic resistance</i>	43
Eduardo J. Izquierdo, Miguel Aguilera and Randall D. Beer. <i>Analysis of Ultrastability in Small Dynamical Recurrent Neural Networks</i>	51
Dov Pechenick, Jason Moore and Joshua Payne. <i>The influence of assortativity on the robustness and evolvability of gene regulatory networks upon gene birth</i>	59
Flavio L. Pinheiro, V��tor V. Vasconcelos, Francisco C. Santos and Jorge M. Pacheco. <i>Self-organized game dynamics in complex networks</i>	61
Stefano Nichele and Gunnart Tufte. <i>Evolution of Incremental Complex Behavior on Cellular Machines</i>	63
Simon Hickinbotham, Michael Weeks and James Austin. <i>The ALife Zoo: cross-browser, platform-agnostic hosting of Artificial Life simulations</i>	71
Micah Rosenkind, David Arnold and Graham Winstanley. <i>A Temporal Pattern Predictor for Virtual Characters</i>	79

Lance Williams. <i>Evolution of Tail-Call Optimization in a Population of Self-Hosting Compilers</i>	86
Qinxin Pan, Ting Hu, Angeline Andrew, Margaret Karagas and Jason Moore. <i>Bladder cancer specific pathway interaction networks</i>	94
Roman Miletitch, Vito Trianni, Alexandre Campo and Marco Dorigo. <i>Information Aggregation Mechanisms in Social Odometry</i>	102
Jie Tan, Jason Moore and Ryan Urbanowicz. <i>Rapid Rule Compaction Strategies for Global Knowledge Discovery in a Supervised Learning Classifier System</i>	110
Christoph Salge, Cornelius Glackin and Daniel Polani. <i>Empowerment and State-dependent Noise - An Intrinsic Motivation for Avoiding Unpredictable Agents</i>	118
Randal Olson, Masoud Mirmomeni, Tim Brom, Eric Brugger, Arend Hintze, David Knoester and Christoph Adami. <i>Evolved digital ecosystems: Dynamic steady state, not optimal fixed point</i>	126
Omer Markovitch and Doron Lancet. <i>Prebiotic Evolution of Molecular Assemblies: From Molecules to Ecology</i>	134
Veljko Potkonjak, Vladimir M. Petrovic, Kosta Jovanovic and Dragan Kostic. <i>Human-Robot Analogy - How Physiology Shapes Human and Robot Motion</i>	136
Alessandro Fontana and Borys Wrobel. <i>An artificial lizard regrows its tail (and more): regeneration of 3-dimensional structures with hundreds of thousands of artificial cells</i>	144
Lenka Pitonakova and Seth Bullock. <i>Controlling Ant-Based Construction</i>	151
Markus Brede and Jason Noble. <i>The coevolution of costly heterogeneities and cooperation in the prisoner's dilemma game</i>	159
Miguel Gonzalez, Richard Watson, Jason Noble and Patrick Doncaster. <i>Multiple Life-History Stage Competition and its Effect on Coexistence</i>	167
Taichi Haruna. <i>Robustness and Directed Structures in Ecological Flow Networks</i>	175
Vitor V. Vasconcelos, Flavio L. Pinheiro, Francisco C. Santos and Jorge M. Pacheco. <i>Bootstrapping back the climate with self-organization</i>	182
Benjamin Inden and Jürgen Jost. <i>Neural agents can evolve to reproduce sequences of arbitrary length</i>	184
Pierre Buesser and Marco Tomassini. <i>Spatial Organisation of Cooperation with Contingent Agent Migration</i>	192
Dusan Misevic, Antoine Frenoy and François Taddei. <i>In silico evolution of transferable genetic elements</i>	200
Daiki Komatsu, Kei Fujiwara and Shin-Ichiro M. Nomura. <i>Construction of a remote-controlled supramolecular micro-crawler</i>	208

Declan Baugh and Barry McMullin. <i>Evolution of G-P mapping in a von Neumann Self-reproducer within Tierra</i>	210
Paul Szerlip and Kenneth Stanley. <i>Indirectly Encoded Sodarace for Artificial Life</i>	218
Alberto Antonioni, Mattia Egloff and Marco Tomassini. <i>An Energy-Based Model for Spatial Social Networks</i>	226
Arash Khabbaz Saberi, Jan Friso Groote and Sarmen Keshishzadeh. <i>Analysis of Path Planning Algorithms: a Formal Verification-based Approach</i>	232
Nathaniel Virgo and Takashi Ikegami. <i>Autocatalysis Before Enzymes: The Emergence of Prebiotic Chain Reactions</i>	240
Matthew Egbert, Gerd Gruenert, Gabi Escuela and Peter Dittrich. <i>Synthetic signalling protocell networks as models of neural computation</i>	248
Tadao Maekawa, Manabu Honda, Norie Kawai, Emi Nishina, Osamu Ueno and Tsutomu Oohashi. <i>Heterogeneity and complexity of a simulated terrestrial environment account for the superiority of the altruistic gene</i>	250
Maja Murnik, Špela Petric and Miha Turšič. <i>Beyond Life Cycles</i>	258
Jean-Marc Montanier and Nicolas Bredeche. <i>Evolution of Altruism and Spatial Dispersion: an Artificial Evolutionary Ecology Approach</i>	260
Paul Williams and Randall Beer. <i>Environmental Feedback Drives Multiple Behaviors from the Same Neural Circuit</i>	268
Zoe Bukovac, Alan Dorin and Adrian Dyer. <i>A-Bees See: A Simulation to Assess Social Bee Visual Attention During Complex Search Tasks</i>	276
Atanas Radenski. <i>Using MapReduce Streaming for Distributed Life Simulation on the Cloud</i>	284
Kai Olav Ellefsen. <i>Balancing the Costs and Benefits of Learning Ability</i>	292
Markus Brede and Simon Tudge. <i>Replication strategies and the evolution of cooperation by exploitation</i>	300
James Snowdon and Ben Waterson. <i>When congestion can be useful: modelling driver diversion behaviour in road traffic networks</i>	308
Tomonori Hasegawa and Barry McMullin. <i>Exploring the Point-mutation Space of a von Neumann Self-reproducer within the Avida World</i>	316
Hiroki Mori, Yuzi Okuyama and Minoru Asada. <i>Emergence of diverse behaviors from interactions between nonlinear oscillator complex networks and a musculoskeletal system</i>	324



Dharani Punithan and Bob McKay. <i>Collective Dynamics and Homeostatic Emergence in Complex Adaptive Ecosystem</i>	332
Christian Darabos, Craig O. Mackenzien, Marco Tomassini, Mario Giacobini and Jason H. Moore. <i>Cellular Automata Coevolution of Update Functions and Topologies: A Tradeoff between Accuracy and Speed</i>	340
Christian Darabos, Britney E. Graham, Ting Hu and Jason H. Moore. <i>Bipartite Networks Show the Genotype-to-Phenotype Relationship in Biological Systems Models: A Study of the Robustness, Evolvability, and Accessibility in Linear Cellular Automata</i>	348
Adam Jackson and Richard Watson. <i>The Effects of Assortment on Population Structuring Traits on the Evolution of Cooperation</i>	356
Dov Pechenick, Joshua Payne and Jason Moore. <i>An alternative route to robustness: The relationship between assortativity, in-components, and characteristic path length in gene regulatory networks</i>	364
Marco Villani, Alessandro Filisetti, Stefano Benedettini, Andrea Roli, David Avra Lane and Roberto Serra. <i>The detection of intermediate-level emergent structures and patterns</i>	372
Elio Tuci, Boris Mitavskiy and Gianpiero Francesca. <i>On the evolution of self-organised role-allocation and role-switching behaviour in swarm robotics: a case study</i>	379
James Dyke and Iain Weaver. <i>Tipping points in Complex Coupled Life-Environment Systems</i>	387
Miguel Aguilera, Ignacio Morer, Xabier E. Barandiaran and Manuel G. Bedia. <i>Quantifying Political Self-Organization in Social Media. Fractal patterns in the Spanish 15M movement on Twitter</i>	395
Orly Stettiner. <i>Biology of Digital Organisms: How Language and tools Constructs Reality</i>	403
Kai Olav Ellefsen. <i>Evolved Sensitive Periods in Learning</i>	409
Jean Sirmai. <i>Autopoiesis Facilitates Self-Reproduction</i>	417
Solvi F. Arnold, Reiji Suzuki and Takaya Arita. <i>Evolution of Social Representation in Neural Networks</i>	425
Radek Hrbacek and Michaela Sikulova. <i>Coevolutionary Cartesian Genetic Programming in FPGA</i>	431
Takashi Ito, Marcin Pilat, Reiji Suzuki and Takaya Arita. <i>Coevolutionary Dynamics Caused by Asymmetries in Predator-Prey and Morphology-Behavior Relationships</i>	439
John Maher. <i>The Influence of Cell Type on Artificial Development</i>	446
Michal Joachimczak, Taras Kowaliw, Rene Doursat and Borys Wrobel. <i>Controlling development and chemotaxis of soft-bodied multicellular animats with the same gene regulatory network</i>	454
Bijan Ranjbar-Sahraei, Katerina Stankova, Karl Tuyls and Gerhard Weiss. <i>Stackelberg-based Coverage Approach in Nonconvex Environments</i>	462

Takahiro Sasaki. <i>Simulating Limited Diversity in Evolution of Influenza</i>	470
Timothy Moran, Markus Brede, Antonella Ianni and Jason Noble. <i>The Origin of Money: An Agent-Based Model</i>	472
Philippe Collard. <i>Beyond the schelling's segregation: is it equivalent to be repulsed by dissimilar rather to be attracted by similar?</i>	480
Shin-Ichiro M. Nomura, Yusuke Sato and Kei Fujiwara. <i>Molecular robotics approach for constructing an artificial cell model</i>	488
Hisashi Murakami and Yukio Gunji. <i>Lévy-like Distribution Shown by Intermittent Search Model with Misunderstanding Switch Pattern</i>	492
Mizuki Oka, Takashi Ikegami, Alex Woodward, Yiqing Zhu and Kazuhiko Kato. <i>Cooperation, Congestion and Chaos in Concurrent Computation</i>	498
John Lones and Lola Cañamero. <i>Epigenetic adaptation in action selection environments with temporal dynamics</i>	505
James Thorniley and Phil Husbands. <i>Hidden information transfer in an autonomous swinging robot</i>	513
Amber Ferger, Wai Fai Lau, Philipp Ross, Wyman Zhao, Hiroki Sayama and Steen Rasmussen. <i>Impact of Personal Fabrication Technology on Social Structure and Wealth Distribution: An Agent-Based Simulation Study</i>	521
Carlos Fernandes, Juan Laredo, Juan Julian Merelo, Carlos Cotta and Agostinho Rosa. <i>Adapting the Bak-Sneppen Model to a Dynamic and Partially Connected Grid of Hierarchical Species</i>	523
Joachim Erdei, Michal Joachimczak and Borys Wrobel. <i>Evolving gene regulatory networks controlling foraging strategies of prey and predators in an artificial ecosystem</i>	531
Gianluca Massera, Tomassino Ferrauto, Onofrio Gigliotta and Stefano Nolfi. <i>FARSA : An Open Software Tool for Embodied Cognitive Science</i>	538
Minami Takato, Shinpei Yamasaki, Shiho Takahama, Junichi Tanida, Ken Saito and Fumio Uchikoba. <i>Insect Type MEMS Micro Robot Controlled by CMOS IC of Hardware Neural Networks</i>	546
Tiago Baptista and Ernesto Costa. <i>Open-Ended Evolution of a Circadian Rhythm</i>	554
Ezequiel Di Mario, Iñaki Navarro and Alcherio Martinoli. <i>The Effect of the Environment in the Synthesis of Robotic Controllers: A Case Study in Multi-Robot Obstacle Avoidance using Distributed Particle Swarm Optimization</i>	561
Martin Stehlik, Adam Saleh, Andriy Stetsko and Vashek Matyas. <i>Multi-Objective Optimization of Intrusion Detection Systems for Wireless Sensor Networks</i>	569

Yuri Lenon Barbosa Nogueira, Carlos Eduardo Fisch de Brito, Creto Augusto Vidal and Joaquim Bento Cavalcante Neto. <i>Evolving Plastic Neuromodulated Networks for Behavior Emergence of Autonomous Virtual Characters</i>	577
Andrea Guarneri, Dario Maggiorini, Laura Anna Ripamonti and Marco Trubian. <i>GOLEM: Generator Of Life Embedded into MMOs</i>	585
Darius Falahat, Enrico Gerding and Markus Brede. <i>Evolving Behaviour-Dependent Strategies in Agent Negotiations</i>	593
Ralf Der and Georg Martius. <i>Behavior as broken symmetry in embodied self-organizing robots</i>	601
Payam Zahadat, Karl Crailsheim and Thomas Schmickl. <i>Social Inhibition Manages Division of Labour in Artificial Swarm Systems</i>	609
Tsubasa Azumagakito, Reiji Suzuki and Takaya Arita. <i>Cyclic Behavior in Gene-Culture Coevolution Mediated by Phenotypic Plasticity in Language</i>	617
Vasthi Alonso Chavez, James G. Dyke and C. Patrick Doncaster. <i>Detecting regime shifts in artificial ecosystems</i>	625
Sue Attwood, Lola Cañamero and Rene Te Boekhorst. <i>SimianWorld - A Study of Social Organisation Using an Artificial Life Model</i>	633
Simón C. Smith and J. Michael Herrmann. <i>Self-Organisation of Generic Policies in Reinforcement Learning</i>	641
António Homem Ferreira and Carlos Martinho. <i>ASAP: an Ant resource Search Algorithm for swarm-like P2P networks</i>	649
Anna Shcherbacheva and Tuomo Kauranne. <i>Swarm dynamics with limited perception - how do mosquitoes recognize a friend?</i>	657
Zbigniew Pliszka and Olgierd Unold. <i>Some Remarks on Dynamics of Binary Chromosomes Population</i>	663
Manuel Heras-Escribano, Jason Noble and Manuel De Pinedo. <i>The only wrong cell is the dead one: On the enactive approach to normativity</i>	665
Evert Haasdijk and Nicolas Bredeche. <i>Controlling Task Distribution in MONEE</i>	671
Francesco Pugliese and Davide Marocco. <i>The relationship between Flocking Behavior and the Emergence of Leadership</i>	679
Brian Hrolenok and Tucker Balch. <i>Learning Schooling Behavior from Observation</i>	686
Hiroataka Osawa and Michita Imai. <i>Evolution of Mutual Trust Protocol in Human-based Multi-Agent Simulation</i>	692



Julien Hubert, Eiko Matsuda and Takashi Ikegami. <i>Hebbian Learning In A Multimodal Environment</i>	698
Dario Maggiorini, Laura Anna Ripamonti and Samuele Panzeri. <i>Follow the Leader: a Scalable Approach for Realistic Group Behavior of Roaming NPCs in MMO Games</i>	706
Tomohiro Harada and Keiki Takadama. <i>Analyzing Program Evolution in Genetic Programming using Asynchronous Evaluation</i>	713
Payam Zahadat, Thomas Schmickl and Karl Crailsheim. <i>Evolution of Spatial Pattern Formation by Autonomous Bio-Inspired Cellular Controllers</i>	721
Takayuki Niizato. <i>Multiplicity of Interpretation in an Asynchronous Updating Rule: Emergence of Collective Cognition</i>	729
Yara Khaluf and Franz Rammig. <i>Task Allocation Strategy for Time-Constrained Tasks in Robots Swarms</i>	737
Margareta Segerståhl. <i>Formal Model of Living Systems</i>	745
Yuri Lenon Barbosa Nogueira, Carlos Eduardo Fisch de Brito, Creto Augusto Vidal and Joaquim Bento Cavalcante Neto. <i>Emergence of Autonomous Behaviors of Virtual Characters through Simulated Reproduction</i>	750
Takeo Kato, Koichiro Sato and Yoshiyuki Matsuoka. <i>Robust Optimization of Adjustable Control Factors Using Particle Swarm Optimization</i>	758
Paolo Arena, Luca Patanè and Roland Strauss. <i>The insect Mushroom Bodies: a paradigm of neural reuse</i>	765
Alessandro Di Nuovo, Davide Marocco, Santo Di Nuovo and Angelo Cangelosi. <i>Ballistic Action Planning in Robotics by means of Artificial Imagery</i>	773
Sjriek Alers, Bijan Ranjbar-Sahraei, Stefan May, Karl Tuyls and Gerhard Weiss. <i>Evaluation of an Experimental Framework for Exploiting Vision in Swarm Robotics</i>	775
Lorenzo Garattoni, Andrea Roli, Matteo Amaducci, Carlo Pinciroli and Mauro Birattari. <i>Boolean network robotics as an intermediate step in the synthesis of finite state machines for robot control</i>	783
Adrien Padirac, Alexandre Baccouche, Teruo Fujii, Andre Estevez-Torres and Yannick Rondelez. <i>Predator prey molecular landscapes</i>	791
Alessandro Filisetti, Alex Graudenzi, Chiara Damiani, Marco Villani and Roberto Serra. <i>The role of backward reactions in a stochastic model of catalytic reaction networks</i>	793
Vincent Ducharme, Richard Egli and Sylvie Jetté. <i>A cardiopulmonary system for a virtual patient</i>	802
Yukio Gunji, Tomoko Sakiyama, Sohei Wakisaka, Naotaka Fujii and Tomoaki Nakamura. <i>Artificial Causal Space-Time</i>	810

Jitka Cejkova, Damian Gorny, Petra Haufova and František Štěpánek. <i>“Artificial spores” - hybrid alginate microcapsules with encapsulated yeast cells</i>	818
Hideaki Matsubayashi, Yutetsu Kuruma and Takuya Ueda. <i>In vitro Synthesis of Membrane Protein Machinery toward the Construction of Artificial Cell</i>	824
Jerzy Maselko, James Pantaleone and Vitaliy Kaminker. <i>Experimental studied on growing chemical organisms</i>	825

### **Adaptive Hardware & Systems and Bioelectronics**

Julio G. Arriaga, George Kossan, Martin L. Cody, Edgar E. Vallejo and Charles E. Taylor. <i>Acoustic sensor arrays for understanding bird communication. Identifying Cassin’s Vireos using SVMs and HMMs</i>	827
--	-----

### **Adaptive Living Material Technologies & Biomimetic Microsystems**

Henderson Cleaves and Neil Anthony. <i>Prebiotic Organic Microstructures as Model Protocells</i>	829
--	-----

### **Artificial Immune, Neural and Endocrine Systems**

Ronald Thenius, Payam Zahadat and Thomas Schmickl. <i>EMANN - a model of emotions in an artificial neural network</i>	830
Stein Keijzers, Peter Maandag, Elena Marchiori and Ida Sprinkhuizen-Kuyper. <i>Image Similarity Search using a Negative Selection Algorithm</i>	838

### **Artificial Immune Systems - ICARIS**

Huikeng Lau, Iain Bate and Jon Timmis. <i>Immune-Inspired Error Detection for Multiple Faulty Robots in Swarm Robotics</i>	846
Grazziela Figueredo, Peer-Olaf Siebers, Douglas Augusto, Helio Barbosa and Uwe Aickelin. <i>The use of Agent-based Simulation to Discover Extreme Cases in Immune-Interactions with Early-Stage Cancer Scenarios</i>	854
Kevin Sim, Emma Hart and Ben Paechter. <i>Learning to Solve Bin Packing Problems with an Immune Inspired Hyper-heuristic</i>	856
Nicola Capodieci, Emma Hart and Giacomo Cabri. <i>An immune network approach for self-adaptive ensembles of autonomic components: a case study in swarm robotics</i>	864
Wilburn W. P. Tsang and Henry Y. K. Lau. <i>An Artificial Immune System-based Many-Objective Optimization Algorithm with Network Activation Scheme</i>	872
Mark Read, Magnus Tripp, Hannah Leonova, Louis Rose and Jon Timmis. <i>Automated calibration of agent-based immunological simulations</i>	874

Shangce Gao, Tao Gong, Weiya Zhong, Fang Wang and Beibei Chen. <i>Enhancing the learning capacity of immunological algorithms: a comprehensive study of learning operators</i>	876
Dongdong Zhao and Wenjian Luo. <i>Real-Valued Negative Databases</i>	884
Emma Hart, Mark Read, Chris McEwan, Uwe Aickelin and Julie Greensmith. <i>On the Role of the AIS Practitioner</i>	891
Paul Taylor, Fiona Polack and Jon Timmis. <i>Accelerating Immunos 99</i>	893
Henry Lau and Nicole Lee. <i>Modelling the migration and maturation of dendritic cells for automatic optimization of complex engineering problems</i>	899
Lingli Yu, Binglu Wu and Tao Gong. <i>A hierarchical support vector machine based on feature-driven method for speech emotion recognition</i>	901

### **Bioinspired Learning and Optimization**

Ana L. C. Bazzan. <i>Comparing Reinforcement Learning and Evolutionary Based Adaptation in Population Games</i>	909
Paulo Urbano and Loukas Georgiou. <i>Improving Grammatical Evolution in Santa Fe Trail using Novelty Search</i>	917
Matthew Crossley, Andy Nisbet and Martyn Amos. <i>Quantifying the Impact of Parameter Tuning on Nature-Inspired Algorithms</i>	925
Arnaud Zinflou, Caroline Gagne and Marc Gravel. <i>A hybrid genetic/immune strategy to tackle the multi-objective quadratic assignment problem</i>	933
Ariadne Costa, Patrícia Vargas and Renato Tinós. <i>Using explicit averaging fitness for studying the behaviour of rats in a maze</i>	940

### **Bioinspired Robotics**

Heiko Hamann. <i>Speciation Dynamics: Generating Selective Pressure Towards Diversity</i>	947
Shuhei Miyashita and Daniela Rus. <i>Multi-crease Self-folding by Uniform Heating</i>	955
Claire E. Gerrard, John McCall, George Coghill and Christopher Macleod. <i>Artificial Reaction Network Agents</i>	957
Mark Khazanov, Ben Humphreys, William Keat and John Rieffel. <i>Exploiting Dynamical Complexity in a Physical Tensegrity Robot to Achieve Locomotion</i>	965
Adam Stanton and Alastair Channon. <i>Heterogeneous complexification strategies robustly outperform homogeneous strategies for incremental evolution</i>	973



Gregory Vorobyev, Andrew Vardy and Wolfgang Banzhaf. <i>Conformity and Nonconformity in Collective Robotics: A Case Study</i>	981
Xiaofeng Xiong, Florentin Woergoetter and Poramate Manoonpong. <i>A Neuromechanical Controller of a Hexapod Robot for Walking on Sponge, Gravel and Snow Surfaces</i>	989
Fabien Hervouet and Eric Bourreau. <i>FIMO: Framework for Intrinsic Motivation</i>	997
Filippo Vella, Ignazio Infantino, Giovanni Pilato and Riccardo Rizzo. <i>A practical approach to Humanoid Introspection</i>	1005
Andrew Philippides, Bart Baddeley, Phil Husbands and Paul Graham. <i>A Situated and Embodied Model of Ant Route Navigation</i>	1007
G. Matthew Fricke, Francois Asperti-Boursin, Joshua Hecker, Judy Cannon and Melanie Moses. <i>From Microbiology to Microcontrollers: Robot Search Patterns Inspired by T Cell Movement</i>	1009
Sebastian Clarke, Frederic Labrosse, Vito Trianni and Elio Tuci. <i>An evolutionary approach to road following: a simulated case study</i>	1017
Joshua Hecker, Karl Stolleis, Bjorn Swenson, Kenneth Letendre and Melanie Moses. <i>Evolving Error Tolerance in Biologically-Inspired iAnt Robots</i>	1025
Christopher Schwarzer and Nico Michiels. <i>Conditions for Outperformance of Recombination in Online Evolution of Swarm Robots</i>	1033
Daniela Kengyel, Ronald Thenius, Karl Crailsheim and Thomas Schmickl. <i>Influence of a Social Gradient on a Swarm of Agents Controlled by the BEECLUST Algorithm</i>	1041
Vincenzo Fioriti, Stefano Chiesa and Fabio Fratichini. <i>Expanding the Time Horizon in Underwater Robot Swarms</i>	1049
A.E. Eiben, Nicolas Bredeche, Mark Hoogendoorn, Juergen Stradner, Jon Timmis, Andy Tyrrell and Alan Winfield. <i>The Triangle of Life</i>	1056

## **Biologically Inspired Engineering**

Martin Ullrich, Petra Haufová, Mandeep Singh and František Štěpánek. <i>Radiofrequency triggered enzymatic reaction inside hydrogel microparticles</i>	1064
--	------

## **Evolvable Hardware, Evolutionary Electronics & BioChips**

Leroy Cronin. <i>Linking Evolution in Silico, Hardware, and Chemistry to discover or engineer Inorganic Biology</i>	1066
Samuel Chapman, David Knoester, Arend Hintze and Christoph Adami. <i>Evolution of an artificial visual cortex for image recognition</i>	1067

Eiko Matsuda, Takeshi Mita, Julien Hubert, Mizuki Oka, Douglas Bakkum, Urs Frey, Hirokazu Takahashi and Takashi Ikegami. <i>Multiple Time Scales Observed in Spontaneously Evolved Neurons on High-density CMOS Electrode Array</i>	1075
---	------

## **Foundations of Complex Systems and Biological Complexity**

Steen Rasmussen, Harold Fellermann and Shinpei Tanaka. <i>Sequence selection and evolution in a binary polymer model</i>	1083
Gonzalo Ruz, Marco Montalva and Eric Goles. <i>On the preservation of limit cycles in Boolean networks under different updating schemes</i>	1085
Jack Hessel and Sherri Goings. <i>Using Reproductive Altruism to Evolve Multicellularity in Digital Organisms</i>	1091
Martin Biehl and Daniel Polani. <i>One way to see two in one</i>	1099

## **Mathematical Models for the Living Systems and Life Sciences**

Paul Calcraft, Phil Husbands and Andrew Philippides. <i>An Environmental Model of Self-Compatibility Transitions in the Solanaceae Plant Family</i>	1107
Jure Demšar and Iztok Lebar Bajec. <i>Family Bird: A Heterogeneous Simulated Flock</i>	1114
Pedro Mariano and Luis Correia. <i>Population Dynamics of Centipede Game using an Energy Based Evolutionary Algorithm</i>	1116
Tobias Meister, Ronald Thenius, Daniela Kengyel and Thomas Schmickl. <i>Cooperation of two different swarms controlled by BEECLUST algorithm</i>	1124

## **Music and the Origins and Evolution of Language**

Jonathan Impett and Leonardo Impett. <i>The perception of potential: interference, dimensionality and knowledge</i>	1126
---	------

## **Programmable Nanomaterials**

Leroy Cronin. <i>Programming the Assembly of Inorganic Nanomaterials Using Networked Chemical Reactions</i>	1132
Darko Stefanovic, Milan Stojanovic, Mark Olah and Oleg Semenov. <i>Catalytic Molecular Walkers: Aspects of Product Release</i>	1134

## ***Synthetic and Systems Biochemistry and Biological Control***

Leroy Cronin. <i>Synthetic biology manipulations in 3D printed wet-ware</i>	1142
Nathanael Aubert, Quang Huy Dinh, Masami Hagiya, Hitoshi Iba, Teruo Fujii, Nicolas Bredeche and Yannick Rondelez. <i>Evolution of Cheating DNA-based Agents Playing the Game of Rock-Paper-Scissors</i>	1143
Christoph Flamm, Chris Hemmingsen and Daniel Merkle. <i>Barrier Trees for Metabolic Adjustment Landscapes</i>	1151
Yoshihiro Shimizu and Yo Tanaka. <i>Reconstitution of the protein synthesis system on a glass microchip</i>	1159
Pasquale Stano, Erica D'Aguanno, Paolo Carrara, Alfred Fahr and Pier Luigi Luisi. <i>Recent advancements in synthetic cells research</i>	1160
Ben Shirt-Ediss, Ricard Solé and Kepa Ruiz-Mirazo. <i>Steady state analysis of a vesicle bioreactor with mechanosensitive channels</i>	1162

## ***Late Breaking Papers***

Inman Harvey. <i>How Fast Can We Evolve Something?</i>	1170
Önder Gürcan. <i>An Emergent Model for Mimicking Human Neuronal Pathways in Silico</i>	1172
Huy Tran and Andre S. Ribeiro. <i>Effects of inducer intake kinetics on the dynamics of gene expression</i>	1174
Cai Li, Robert Lowe and Tom Ziemke. <i>Crawling Posture Learning in Humanoid Robots using a Natural-Actor-Critic CPG Architecture</i>	1182
Ignazio Infantino and Riccardo Rizzo. <i>An Artificial Behavioral Immune System for cognitive robotics</i>	1191
Michael Spranger. <i>Evolutionary Explanations for Spatial Language - A Case Study on Landmarks</i>	1199
Alaa Abi Haidar, Adrien Six, Jean-Gabriel Ganascia and Veronique Thomas-Vaslin. <i>The Artificial Immune Systems Domain: Identifying Progress and Main Contributors Using Publication and Co-Authorship Analyses</i>	1206
Geoff Nitschke and Olaf Witkowski. <i>The Transmission of Migratory Behaviors</i>	1218
Genki Ichinose, Masaya Saito, Hiroki Sayama and David Sloan Wilson. <i>Adaptive long-range migration promotes cooperation under tempting conditions</i>	1220



Hector Zenil, Gordon Ball and Jesper Tegner. <i>Testing Biological Models for Non-linear Sensitivity with a Programmability Test</i>	1222
Chris Gordon-Smith. <i>SimSoup: Molecules Designed for Switchable Autocatalytic Memory</i>	1224
Roland Cazalis. <i>The emergence of life and evolution: Towards a categorial approach</i>	1232
Yao Yao, Kathleen Marchal and Yves Van de Peer. <i>Using novel bio-inspired principles to improve adaptability of evolutionary robots in dynamically changing environments</i>	1234
Peter Scully, Jingping Song, Jules Pagna Disso and Mark Neal. <i>CARDINAL-E: AIS Extensions to CARDINAL for Decentralised Self-Organisation for Network Security</i>	1235
<b>Author List</b>	1237

## ECAL 2013 Preface

ECAL 2013 continues with its mission to play a unique role as an information and knowledge sharing forum in synthetic life, where the wide range of its offerings allows ECAL attendants to learn and network.

After a pioneering phase lasted more than twenty years, ECAL will show the current state of the art of a mature and autonomous discipline, Artificial Life, that is collocated at the intersection between a theoretical perspective, namely the scientific explanations of different levels of life organizations (e.g., molecules, compartments, cells, tissues, organs, organisms, societies, collective and social phenomena), and advanced technological applications (bio-inspired algorithms and techniques to building-up concrete solutions such as in robotics, data analysis, search engines, gaming).

ECAL 2013, the twelfth European Conference on Artificial Life, is held in Taormina, on the beautiful island of Sicily, Italy, in September 2-6, 2013. While hosting the event, the city of Taormina will give the participants the opportunity to enjoy the richness of its historical and cultural atmosphere, its traditions, and the beauty of its natural resources, the sea, and Mount Etna the largest and active volcano of Europe (a UNESCO World Heritage site).

The 12th ECAL is truly “grand celebration” with hundreds of paper and poster presentations, *five focused tutorials* and an impressive constellation of *ten satellite workshops*. The scientific program has been designed to optimize the interactions on all levels. This year’s program includes papers from one of the largest pool of submissions (*267 submissions*).

Moreover, at ECAL 2013 we added *new and exciting tracks*: Adaptive Hardware & Systems and Bioelectronics, Adaptive Living Material Technologies & Biomimetic Microsystems, Artificial Immune, Neural and Endocrine Systems, Artificial Immune Systems - ICARIS, Bioinspired Learning and Optimization, Bioinspired Robotics, Biologically Inspired Engineering, Evolvable Hardware, Evolutionary Electronics & BioChips, Foundations of Complex Systems and Biological Complexity, Mathematical Models for the Living Systems and Life Sciences, Music and the Origins and Evolution of Language, Programmable Nanomaterials, and Synthetic and Systems Biochemistry and Biological Control.

So, in the ECAL 2013 program you will find research works written by leading scientists in the field, from *fifty different countries and five continents*, describing an impressive array of results, ideas, technologies and applications. The keynotes have always been one of the most important parts of ECAL. The nine keynote speakers of ECAL 2013 will focus on a wide spectrum of topics of our scientific and technological ecosystem. In particular, the nine keynote speakers are the following:

- *Roberto Cingolani*, Italian Institute of Technology - IIT, Italy
- *Roberto Cipolla*, University of Cambridge, UK
- *Dario Floreano*, Ecole Polytechnique Fédérale de Lausanne - EPFL, Switzerland
- *Martin Hanczyc*, University of Southern Denmark, Denmark
- *Henrik Hautop Lund*, Technical University of Denmark, Denmark
- *Keymeulen*, California Institute of Technology - CALTECH, USA
- *Steve Oliver*, University of Cambridge, UK
- *Bernhard Palsson*, University of California San Diego - UCSD, USA
- *Rolf Pfeifer*, Swiss Federal Institute of Technology - ETH, Switzerland

These speakers make a blend of all the Artificial Life topics, in particular their choice represents one of the first cross talk between synthetic (or systems biology) and robotics through the concept of artificial life. We expect both the round table with the speakers and the frequent non-formal interactions with the researchers attending the conference and the workshops will represent remarkable events!

This edition has highlighted a more profound integration of concepts and ideas from life sciences, artificial intelligence, mathematics, engineering and computer science than in the past. Furthermore, the integration between dry and wet lab biological results shows more progress. Although synthetic biology appears on a small number of

papers, it is already showing itself as a powerful engine for boosting new ideas into the ECAL breath of topics and new type of researchers, perhaps interested in designing life at different levels of complexity, will follow ECAL or will consider ECAL as top conference. As organisers of such important event we felt the duty to ask ourselves three questions:

Will this event attract young inquisitive minds?

Will this event be full of opportunity and career boosts for established researchers in the artificial life fields?

Will an ECAL author, or a student attending it, change the world?

We have shaped the conference to answer all these questions. We believe this conference is the place for rapid exchange of very innovative ideas in artificial life and therefore has a very important role in the current geography of places where innovation could take place. A young researcher will be exposed to the largest diversity of ideas in artificial life. The expectations are reflected by the larger number of registrations, papers, and satellite workshops with respect to the previous editions.

Finally, we would like to recognize the enormous efforts of the ECAL organizing committee who made ECAL possible by donating their time, expertise, and enthusiasm. Without their hard work and dedication, ECAL would not be possible. We also could not have organized ECAL 2013 without the excellent work of all of the program committee members, our workshop chair, tutorial chair, publicity chair, financial manager, conference secretary and local organizers. We would like to express our appreciation to the plenary speakers, to the tutorial speakers, to the workshop organizers, and to all the authors who submitted research papers to ECAL 2013.

ECAL is the premier event for science and technology in synthetic life, where scientists from all over the world meet to exchange ideas and sharpen their skills.

Taormina, September 2013

Pietro Liò, Orazio Miglino, Giuseppe Nicosia, Stefano Nolfi, and Mario Pavone

## Keynote Speakers

### Nanotechnologies for Humans and Humanoids

- **Roberto Cingolani** is the Scientific Director of IIT since December 8th, 2005. He earned a Ph.D. in Physics from the University of Bari in 1988. From 1989 to 1991, he was a staff member at Max Planck Institute for Festkörperforschung in Stuttgart (Germany). Since 2001, he has been member of various panels of the European Commission within the Sixth and Seventh Frame Programs in the field of Nanotechnology, New materials and New production systems. From 2002, he has been member of different panels of the Ministry of Research and University (Technical Secretariat for the National Research Plan, Commission for the selection of the Centres of Excellence). From 2000 to 2003, he was executive Vice-president of the National Institute for the Physics of the Matter (INFM). Founder and Director of the National Nanotechnology Laboratory (NNL) of INFM at University of Lecce in 2001, R. Cingolani is author or co-author of about 700 papers in international journals and holds about 30 patents in the fields of structural, optical and electronic properties of quantum nanostructures of semiconductors, molecular nanotechnologies for plastic photonics, OLED and plastic electronic devices (since 2000), bio-nanotechnologies, biomimetic systems, biological electronic devices (since 2003) and smart nanocomposite materials.

**Abstract** Nanotechnology is developing along a pathway which is parallel to that of evolution. Nanocomposite biomimetic materials, new sensing devices, interconnection of living cells (organs) and circuits are boosting the development of complex integrated systems such as humanoids and animaloids, whose performances, either biomechanic or cognitive, are continuously improving. A number of new technologies is thus first developed for these advanced machines, and then transferred to humans. Following the concept of the evolutionary pathway of technology we will briefly describe a few representative examples developed at IIT over the last few years:

- Artificial molecules and artificial antibodies, and their application to drug delivery and diagnostics
- Plantoids and robots with sensing roots
- Animaloid (quadrupeds) with advanced equilibrium and motion characteristics and their application to disaster recovery
- Humanoids with unprecedented cognitive and biomechanic capabilities, and their application as human companions, and for rehabilitation and prosthetic tools
- New biocompatible materials for soft machines

### Computer Vision: Making Machines that See

- **Roberto Cipolla** is a Professor of Information Engineering at the University of Cambridge and the Director of Toshiba's (Toshiba Research Europe) Cambridge Research Laboratory. He obtained a B.A. (Engineering) from the University of Cambridge in 1984 and an M.S.E. (Electrical Engineering) from the University of Pennsylvania in 1985. From 1985 to 1988 he studied and worked in Japan at the Osaka University of Foreign Studies (Japanese Language) and Electrotechnical Laboratory. In 1991 he was awarded a D.Phil. (Computer Vision) from the University of Oxford and from 1991-92 was a Toshiba Fellow and engineer at the Toshiba Corporation Research and Development Centre in Kawasaki, Japan. He joined the Department of Engineering, University of Cambridge in 1992 as a Lecturer and a Fellow of Jesus College. He became a Reader in Information Engineering in 1997 and a Professor in 2000. His research interests are in computer vision and robotics and include the recovery of motion and 3D shape of visible surfaces from image sequences; object detection and recognition; novel man-machine interfaces using hand, face and body gestures; real-time visual tracking for localisation and robot guidance; applications of computer vision in mobile phones, visual inspection and image-retrieval and video search. He has authored 3 books, edited 8 volumes and co-authored more than 300 papers.

**Abstract** Computer vision is the science and technology of making machines that see. The talk will begin with an overview of the state-of-the-art in the 3R's of computer vision: registration, reconstruction and

recognition and will include demonstrations of research projects from the University of Cambridge. and Toshiba Research Europe's Cambridge Laboratory.

The second part of the talk will introduce a novel digital interface - a talking head created by Toshiba Research Europe and the University of Cambridge. We have developed a system that can generate a realistic expressive talking head animation. The user enters input text and selects an expression such as 'happy' and 'angry' and the software makes a previously recorded face model talk at an unprecedented level of realism.

The face and speech model was learned from a large training dataset where sentences are spoken with a number of different emotions. In addition to a neutral style, the corpus includes angry, happy, sad, tender and fearful expressions. The realism of the animation is achieved by novel training and face modelling algorithms. A key technology behind training the expressive "talking head" model is Cluster Adaptive Training (CAT), which allows flexible control over the expressiveness of both the voice and the face model.

The new technology allows next generation interfaces. By combining speech and face video synthesis, so-called visual speech synthesis, interaction with computers will become more similar to interacting with another person. A demonstration will be included at the end of the talk.

## Bio-mimetic Flying Robots

- **Dario Floreano** is full professor, Director of the Laboratory of Intelligent Systems at Ecole Polytechnique Fédérale de Lausanne Switzerland (EPFL) and Director of the Swiss National Center of Robotics, a national strategic initiative bringing together all major robotics labs in Switzerland. His research focuses on the convergence of biology, artificial intelligence, and robotics. He has published more than 300 peer-reviewed papers, which have been cited more than 9K times, and four books on the topics of evolutionary robotics, bio-inspired artificial intelligence, and bio-mimetic flying robots with MIT Press and Springer Verlag. He is member of the World Economic Forum Council on robotics and smart devices, co-founder of the International Society of Artificial Life, Inc. (USA), co-founder of the aerial robot company senseFly Ltd (now member of the Parrot Group), advisor to the European Commission for Future Emerging Technologies, member of the editorial board of 10 professional journals, and board member of numerous professional societies in robotics and artificial intelligence. He is also active in the public understanding of robotics and artificial intelligence, delivered almost 150 invited talks worldwide, and started the popular robotics podcast Talking Robots (now The RobotsPodcast).

**Abstract** I will present an overview of my lab's efforts to develop autonomous robots capable of flying in cluttered environments and in safe interaction with humans. I will start by presenting miniature and small-size robots capable of performing collision-free flight and altitude control indoor and outdoor by means of insect-inspired vision and control. I will also present evolved and bio-mimetic strategies for coordination of outdoor flying robots. Finally, I will revisit the conventional concept of flying robots and describe recent work on the development of flying robots capable of surviving and exploiting collisions, just like insects do, in order to explore semi-collapsed buildings or extremely cluttered environments with no light.

## The real artificial lives of droplets

- **Martin Hanczyc** is an Associate Professor at the Institute of Physics and Chemistry at the University of Southern Denmark. He formally was an Honorary Senior Lecturer at the Bartlett School of Architecture, University College London and Chief Chemist at ProtoLife. He received a bachelor's degree in Biology from Pennsylvania State University, a doctorate in Genetics from Yale University and was a postdoctorate fellow under Jack Szostak at Harvard University. He has published in the area of protocells, complex systems, evolution and the origin of life in specialized journals including JACS and Langmuir as well as PNAS and Science. He is also a mentor for the first iGEM synthetic biology student team from Denmark. He is developing novel synthetic chemical systems based on the properties of living systems. Martin actively develops outreach for his research by giving several public lectures and collaborating with architects and artists in several exhibitions

world wide including the Architecture Biennale in Venice Italy in 2010 to bring experiments out of the lab and into the public space. His approach to science has been integrative, multidisciplinary and publicly visible with over 20 press items including Nature News, Scientific American, Discovery Channel, and BBC Radio. Martin gave an invited public lecture at TED in 2011, which now has over 500,000 views.

**Abstract** My work is focused on understanding the fundamental principles of living and evolving systems through experimental science. To this end, I build synthetic systems where dynamic life-like properties emerge when self-assembled systems are pushed away from equilibrium. I will present an experimental model of bottom-up synthetic biology: chemically-active oil droplets. This system has the ability to sense, metabolize and the potential to evolve. Specifically, I will present how sensory-motor coupling can produce chemotactic motile droplets and may form the basis for intelligent and self-replicating materials. In addition, I am involved with a new consortium to develop a robotic interface with feedback to maintain and manipulate the non-equilibrium state of the chemical systems in real time. This represents the integration of chemical, computational, and robotic artificial life.

## Playware ABC

- **Henrik Hautop Lund**, Technical University of Denmark, is head of the Center for Playware. He is World Champion in RoboCup Humanoids Freestyle 2002, has developed shape-shifting modular robots, and has collaborated closely on robotics, ALife and AI with companies like LEGO, Kompan, BandaiNamco, etc. for the past two decades. His Center for Playware at the Technical University of Denmark has a long track record of developing modular playware for playful contextualized IT training in Sub-Saharan Africa and for playful rehabilitation of mentally and physically handicapped children and adult in rural areas of Sub-Saharan Africa. These modular playware technology developments include I-Blocks (LEGO bricks with processing power) and modular interactive tiles (larger bricks for physical rehab). Further, with the development of East-Africa's first science and business park, local entrepreneurship has been fostered amongst students graduating from the university degree programs in contextualized IT. Combining such skills, it became possible to develop technical skill enhancing football games and global connectivity based on modular playware for townships in South Africa for the FIFA World Cup 2010. Lately, together with international pop star and World music promoter Peter Gabriel, it has been possible to develop the MusicTiles app as a music 2.0 experience to enhance music creativity amongst everybody, even people with no initial musical skills whatsoever. In all cases, the modular playware technology approach is used in a playful way to enhance learning and creativity amongst anybody, anywhere, anytime.

**Abstract** Embodied Artificial Life research has led to the development of playware defined as intelligent hardware and software that creates play and playful experiences for users of all ages. With recent technology development, we become able to exploit robotics, modern artificial intelligence and embodied artificial life to create playware which acts as a play force that inspires and motivates users to enter into a play dynamics. In such play dynamics, users forget about time and place, and simultaneously increase their creative, cognitive, physical, and social skills. The Playware ABC concept allows you to develop solutions for anybody, anywhere, anytime through building bodies and brains to allow people to construct, combine and create. Designing playware technology that results in specific behaviors of the user is not a trivial task, and it demands an array of background knowledge in a number of scientific fields. Indeed, definition of desired interactions and behaviors should arise from deep knowledge of the field of application (e.g. play of a specific user group, clinical knowledge of therapy of a specific patient group, professional music knowledge, and professional sport knowledge). In order to meet a practice, where several disciplines can join to develop such playware, and inspired by early artificial life work, we conceptualized the approach of modular playware in the form of building blocks. Building blocks should allow easy and fast expert-driven or user-driven development of playware applications for a given application field. The development of such modular playware technology takes its inspiration from modular robotics, human-robot interaction and embodied artificial life. In this talk, I will present the design principles for creating such modular playware technology with focus on the embodied AI principles that forms the foundation for the design principles of modular playware technology. I will exemplify the design principles with practical applications from the fields of play, sports, music, performance art, and health.

## Self-Repairing and Tuning Reconfigurable Electronics for Space

- **Didier Keymeulen** joined the computer science division of the Japanese National Electrotechnical Laboratory as senior researcher in 1996. In 1998 he moved to the California Institute of Technology and is currently principal member of the technical staff in the Bio-Inspired Technologies Group. He is leading several research tasks on adaptive computing, fault-tolerant electronics, and autonomous and adaptive sensor technologies. He was the electronics test lead of the tunable laser spectrometer (TLS) instrument on the 2011 Mars Science Laboratory (MSL) rover mission to Mars. He serves as chair, co-chair, and program-chair of the NASA/ESA Annual Conferences on Adaptive Hardware and Systems. Didier received his BSEE, MSEE and Ph.D. in Electrical Engineering and Computer Science from the Free University of Brussels, Belgium.

**Abstract** Space missions often require technologies not yet available for earth applications. This talk will present the development of self-reconfigurable electronics for few real-world problems encountered in space applications: survival in extreme environment, high precision inertial measurement for navigation, and in-situ adaptive control for space instruments. Radiation and extreme-temperature hardened electronics is needed to survive the harsh environments beyond earth's atmosphere. Traditional approaches to preserve electronics incorporate radiation shielding, insulation and redundancy at the expense of power and weight. This presentation will demonstrate the implementation of a self-adaptive system using a field programmable gate array (FPGA) and data converters which can autonomously recover the lost functionality of a reconfigurable analog array (RAA) integrated circuit (IC). The second application is related to the development of inexpensive, navigation grade, miniaturized inertial measurement unit (IMU), which surpasses the state-of-the art in performance, compactness (both size and mass) and power efficiency used by current space missions. The talk will explain a self-tuning method for reconfigurable Micro-Electro-Mechanical Systems (MEMS) gyroscopes based on evolutionary computation that has the capacity to efficiently increase the sensitivity of MEMS gyroscopes through in-situ tuning. Finally, we will address the path forward of using adaptive electronics for space.

## The Robot Scientist: Artificial Life Investigates Real Life

- **Steve Oliver** started to work on yeast as a graduate student and has studied it ever since, with occasional excursions into the filamentous fungi and even *Streptomyces* bacteria. The yeast genome-sequencing project was initiated in his lab in the mid-1980's when he started to sequence chromosome III. This turned into a major European Project, which eventually led to the sequencing of the entire yeast genome. He then took up the challenge presented by all the genes of unknown function revealed by the genome sequence, leading the EUROFAN Consortium that pioneered many of the 'omic and other high-throughput technologies in current use. His lab is dedicated to unravelling the workings of the yeast cell, using both top-down and bottom-up systems biology strategies. He is also concerned with developing yeasts as systems to both understand and combat human diseases, including through the use of automated ("Robot Scientist" methods in collaboration with Ross King's group in Aberystwyth). Finally, he takes an interest, at both the bioinformatic and experimental levels, in the evolution of genomes and networks, and is starting to apply this to mammalian systems. The models and experimental systems he uses with yeast sometimes lead in unexpected directions, such as predicting the impact of gene copy number variation in cancer, constructing network models to identify genes important in Alzheimer's Disease, or using yeast "surrogates" to screen for drugs against parasitic diseases .

**Abstract** Science involves the generation of hypotheses and the testing of those hypotheses by experiments whose results are recorded in sufficient detail to enable reproducibility. We developed the Robot Scientist "Adam" to advance the automation of both these processes. Adam has autonomously generated functional genomics hypotheses about the yeast *Saccharomyces cerevisiae*, and experimentally tested those hypotheses using laboratory automation. We, and others, have manually confirmed Adam's conclusions using additional experiments. To describe Adam's experiments we developed an ontology and logical language. The resulting formalisation involves over 10,000 different research units in a nested tree-like structure, ten levels deep, that relates the 6.6 million biomass measurements to their logical description. This formalisation describes how a machine discovered new scientific knowledge. We have now developed a second Robot

Scientist, “Eve”. Like Adam, Eve is a laboratory automation system that uses artificial intelligence techniques to discover scientific knowledge through cycles of experimentation. Eve automates the screening of candidate drugs, hit confirmation, and lead generation through QSAR learning and testing. Econometric modelling has identified the conditions where Eve outperforms standard automation. The second advance is the development of assays based on cellular analog computers. These utilize *Saccharomyces cerevisiae* synthetic biology to compute arbitrary Boolean functions of compound properties. These advances have enabled us to reposition multiple compounds as drugs likely to be effective at inhibiting specific enzyme targets in parasites causing tropical diseases.

## An Insight into Metabolic Requirements of Life

- **Bernhard Ø. Palsson** earned a Ph.D. from the University of Wisconsin in 1984. He held a faculty position at the University of Michigan from 1984 to 1995. He has been with UCSD since 1995. He is the author of over 350 peer reviewed scientific articles. He co-authored the text *TISSUE ENGINEERING*, Prentice Hall in 2004, and wrote *SYSTEMS BIOLOGY: properties of reconstructed networks*, Cambridge University Press in 2006, and *SYSTEMS BIOLOGY: simulation of dynamic network states*, Cambridge University Press in 2011. He sits on the editorial boards of several biology, bioengineering and biotechnology journals. Professor Palsson current research at UCSD focuses on 1) the reconstruction of genome-scale biochemical reaction networks (metabolism, transcriptional regulation & signaling), 2) the development of mathematical analysis procedures for genome-scale models (constraint-based and dynamic models), and 3) the experimental verification of genome-scale models with current emphasis on cellular metabolism and transcriptional regulation in *E. coli*, human pathogens, and organisms that are environmentally & bioprocess importance. He received an Institute of International Education Fellowship in 1977, Rotary Fellowship in 1979, a NATO fellowship in 1984, was named the G.G. Brown Associate Professor at Michigan in 1989, a Fulbright Fellow in 1995, an Ib Henriksen Fellow in 1996, the Olaf Hougen Professorship at the University of Wisconsin in 1999, the Lindbergh Tissue Engineering award in 2001, was named the Galetti Chair of Bioengineering in 2004, was elected into the National Academy of Engineering in 2006, received the UCSD Chancellor’s Associates award in Science and Technology in 2006, and was selected as the developer of one of the most influential technologies on Biotech over the past 10 years by Nature Biotechnology (March 2006). He was the Richard S.H. Mah Lecturer at Northwestern University in 2007, received the Ernst W. Bertner Memorial Award, from the MD Anderson in Houston in 2008, an honorary doctorate from Chalmers University in Gothenburg, Sweden, in 2009, the Marvin Johnson Award from the ACS in 2010, elected fellow of the AAAS on 2011, and received the ASM Promega Biotechnology Research Award in 2012. Professor Palsson is an inventor with over 35 U.S. patents, many of which are in the area of hematopoietic stem cell transplantation, cell culture technology, bioreactor design, gene transfer, cell separations, high-throughput single cell manipulation, pedigree-controlled drug screening, network reconstruction, laboratory adaptive evolution, in silico model building and metabolic engineering. He co-founded a biotechnology company, AASTROM BIOSCIENCES (NASDAQ: ASTM) in 1988, where he served as the Vice President of Developmental Research for two years. Dr. Palsson is the founder and co-founder of ONCOSIS, a company that was focused on the purging of occult tumor cells in autologous bone marrow transplants, renamed as CYNTELLECT, focusing on instrumentation for high-throughput screening and in situ cell sorting and processing, GENOMATICA, a company that is focused on the production of commodity chemicals by fermentation (a spin-off from UCSD), and GT LIFE SCIENCES, an in silico biology (a spin-off from Genomatica).

**Abstract** Whole genome sequencing has enabled us to understand the basic gene portfolio of living cells. A class of gene products that are well known are metabolic enzymes. Based on genome annotation and legacy data it has become possible to reconstruct metabolic networks. These networks are amenable to modeling as systems and have given the basis for in silico cells that are the best representation of their living counterparts. We will discuss the conceptual basis for this field, the difficult and laborious process of network reconstruction, and give examples of the use of in silico cell simulations.



## “Soft Robotics” - the next generation of intelligent machines

- **Rolf Pfeifer** Master’s degree in physics and mathematics and Ph.D. in computer science (1979) from the Swiss Federal Institute of Technology (ETH) in Zurich, Switzerland. Three years as a post-doctoral fellow at Carnegie-Mellon and at Yale University in the US. Since 1987: professor of computer science at the Department of Informatics, University of Zurich, and director of the Artificial Intelligence Laboratory. Visiting professor and research fellow at the Free University of Brussels, the MIT Artificial Intelligence Laboratory in Cambridge, Mass., the Neurosciences Institute (NSI) in San Diego, the Beijing Open Laboratory for Cognitive Science, and the Sony Computer Science Laboratory in Paris. Elected “21st Century COE Professor, Information Science and Technology” at the University of Tokyo in 2004. In 2009: visiting professor at the Scuola Superiore Sant’Anna in Pisa, and at Shanghai Jiao Tong University in China; appointed “Fellow of the School of Engineering” at the University of Tokyo. Currently: Deputy Director of the NCCR Robotics, the “National Competence Center for Research in Robotics” in Switzerland. Research interests: embodied intelligence, biorobotics, morphological computation, modular robotics, self-assembly and educational technology. Authored books: “Understanding Intelligence”, MIT Press, 1999 (with C. Scheier), “How the body shapes the way we think: a new view of intelligence,” 2007 (with Josh Bongard) MIT Press (popular science style), “Designing intelligence - why brains aren’t enough” (short version - with Josh Bongard and Don Berry, e-book), and “La révolution de l’intelligence du corps”, 2012 (“The revolution of embodied intelligence”; with Alexandre Pitti) (in French). Lecture series: “The Shanghai Lectures”, a global mixed-reality lecture series on embodied intelligence, broadcast in 2012 from the University of Zurich, and Shanghai Jiao Tong University, China in cooperation with other universities from around the globe. World exhibition: ROBOTS ON TOUR - World Congress and Exhibition of Robots, Humanoids, Cyborgs, and more. 9 March 2013, Zurich (Puls 5): robotsontour.org.

**Abstract** Researchers from robotics and artificial intelligence increasingly agree that ideas from biology and self-organization can strongly benefit the design of autonomous robots. Biological organisms have evolved to perform and survive in a world characterized by rapid changes, high uncertainty, indefinite richness, and limited availability of information. The term “Soft Robotics” designates a new generation of robots capable of functioning in the real world by capitalizing on “soft” designs at various levels: surface (skin), movement mechanisms (muscles, tendons), and interaction with other agents (smooth, friendly interaction). Industrial robots, in contrast, operate in highly controlled environments with no or very little uncertainty. By “outsourcing” functionality to morphological and material characteristics - e.g. to the elasticity of the muscle-tendon system - the distinction between control and to-be-controlled, which is at the heart of manufacturing and control theory, breaks down and entirely new concepts will be required. In this lecture I will argue that the next generation of intelligent machines - robots - will be of the “soft” kind and I will explore the theoretical and practical implications, whose importance can hardly be over-estimated. I will be using many examples and case studies. In particular I will be introducing the tendon-driven “soft” robot “Roboy” that we have been developing in our laboratory over the last few months. Although many challenges remain, concepts from biologically inspired “soft” robotics will eventually enable researchers to engineer machines for the real world that possess at least some of the desirable properties of biological organisms, such as adaptivity, robustness, and versatility.

## Tutorials

- **Cell Pathway Design for Biotechnology and Synthetic Biology**

Claudio Angione, Jole Costanza, Giovanni Carapezza, Pietro Lió and Giuseppe Nicosia

*Description.* We will introduce the BioCAD framework that we have developed to analyse, optimise and re-design biological models. The framework includes 1) Multi-Objective Optimisation, 2) Sensitivity, 3) Identifiability and 4) Robustness analyses. More specifically, we will present single- and multi-objective optimization algorithms able to handle genetic strategies or uptake rates in a given model. We will show that the condition of Pareto optimality can be relaxed (e.g., epsilon-dominance) to include suboptimal points that can be used to boost the algorithm in its convergence process. The Sensitivity Analysis (SA) is used to compute an index for each parameter that indicates its influence in the model. The Identifiability Analysis (IA) detects functional relations among decision variables through a statistical analysis on the values after and before the optimisation. The Robustness Analysis (RA), Local, Global and Glocal robustness, proves useful to assess the fragileness and robustness of the Pareto optimal solution (or of a given feasible solution) as a result of a perturbation occurring in the model. Our methodology is suitable for (i) any model consisting of ordinary differential equations, differential algebraic equations, flux balance analysis and gene-protein reaction mappings and for (ii) any simulator (e.g., SBML, MatLab, NEURON, C/C++ program). In the tutorial, we will show how these techniques offer avenues to systematically explore, analyse, optimise, design and cross-compare biological models (e.g., metabolic models, gene regulatory networks).

- **Exploring Prebiotic Chemistry Spaces**

Jakob L. Andersen, Christoph Flamm, and Daniel Merkle

*Description.* We have developed a graph grammar based formalism to model chemical transformations. Within our formalism molecules are treated as vertex and edge labeled graphs and reactions (between molecules) are handled as graph rewrite. This approach nicely captures the algebraic properties of real chemistry, where novel molecules can be produced during chemical reactions. Graph grammars, i.e. a set of reaction rules and starting molecules, are very compact representations of entire chemical space. These spaces can contain interesting chemical transformation patterns such as auto-catalytic sub-networks, or alternative routes to molecules of interest. Such sub-networks are usually hard to find due to the vastness of chemical spaces. The situation is especially bad in the origin of life realm, where several putative prebiotic chemistries, all combinatorial complex in nature, have been suggested. Efficient computational methods for constructing and exploring chemical spaces are therefore essential to explore alternative scenarios, or to shade light on potential chemical processes which could have resulted in the emergence of life. The tutorial will offer a mix between short background presentations and accompanying practical examples. To ensure that attendees have the right libraries and programs available, we will provide a working environment. The attendees will learn (i) how to translate chemical reactions to graph rewrite rules, (ii) various methods to explicitly construct chemical spaces (iii) query the chemical space for interesting sub-networks.

- **Designing Adaptive Humanoid Robots Through the FARSA Open-Source Framework**

Gianluca Massera, Tomassino Ferrauto, Onofrio Gigliotta and Stefano Nolfi

*Description.* In this tutorial we will illustrate FARSA, an open-source tool available from <http://laral.istc.cnr.it/farsa/>, that allows to carry on research on Adaptive Robotics. Farsa allows to simulate different robotic platforms (the iCub humanoid robot, and the Khepera, e-Puck, and marXbot wheeled robots), design the sensorimotor system of the robots, design the environment in which the robots operate, perform collective experiments with many interacting robots, design the robots' neural controllers, and allow the robots to develop their behavioural skills through an evolutionary or learning process. It is a cross-platform framework, that works on Linux, Windows and Mac on both 32bit and 64bit systems, constituted by a collection of integrated open-source object-oriented C++ libraries. The framework comes with a powerful graphical application that allow to create and run a large variety of experiments and to analyse and test the obtained results. Furthermore, FARSA has a plugin mechanism that allow to add new features (new robots, new motors, new neural networks, new learning algorithms, etc) that are integrated and accessible

by the graphic interface without modifying and recompiling the core code. FARSA is well documented, easy to use and comes with a series of exemplificative experiments that allow users to quickly gain a comprehension of the tool and a base for running a large spectrum of new experiments that can be set up simply by changing the available parameters. The aim of the tutorial is that to allow also non-technical user to quickly acquire the knowledge required to use the tool and personalize it to specific research interests.

- **Next Generation Sequencing Data Production, Analysis, and Archiving**

Heiko Muller and Luca Zammataro

*Description.* Application of Next Generation Sequencing (NGS) in cancer research is becoming routine in laboratories all over the world and new applications of NGS are being developed at increasing speed. The generation, analysis, interpretation, and storage of NGS data poses a number of technical challenges. Here, the computational infrastructure and the analysis pipelines used at the Center of Genomic Science in Milan (Italian Institute of Technology) are described. In the second part, meta-analysis approaches facilitating the interpretation of NGS data are being discussed. In particular, we will highlight international efforts in cancer genomics aimed at collecting genomic data (e.g. somatic mutations, gene expression, epigenetic modifications, copy number variation) from cancer samples and correlating these data with clinical parameters with the aim of identifying novel biomarkers of cancer subtypes and eventually novel targets for therapeutic intervention. The joined analysis of genomic data of various kinds is a field of active research that is often referred to as Integromics. We will provide an overview of the current state of the art and illustrate the use of selected novel bioinformatic resources of general interest.

- **PyCX: A Python-Based Simulation Code Repository for Complex Systems Education**

Hiroki Sayama

*Description.* This tutorial will introduce PyCX, an online repository of sample codes, all written in plain Python, of various complex systems simulation, including iterative maps, cellular automata, dynamical networks and agent-based models. These sample codes are designed as educational materials so that students can gain practical skills for both complex systems simulation and computer programming simultaneously. The target audience of this tutorial will primarily be educators and researchers who teach complex systems-related courses and thus need simple, easy-to-understand examples of complex systems simulation. The tutorial will also be helpful for students who want to learn basics of writing complex systems simulation themselves. Prior knowledge of Python is helpful but not required. Participants should bring their own laptops to the tutorial so they can work on hands-on coding activities.

## Satellite Workshops

- *A TRUCE workshop on Unconventional Computing in 2070*  
Martyn Amos
- *Artificial Life Based Models of Higher Cognition*  
Onofrio Gigliotta and Davide Marocco
- *Artificial Life in Massive Data Flow*  
Takashi Ikegami, Mizuki Oka, Norman Packard, Mark Bedau and Rolf Pfeifer
- *Collective Behaviours and Social Dynamics*  
Stefano Nolfi, Marco Dorigo, Francesco Mondada, Tom Wenseleers, Vito Trianni and Michael Spranger
- *2nd International Workshop on the Evolution of Physical Systems*  
John Rieffel, Nicolas Bredeche, Jean-Baptiste Mouret and Evert Haasdijk
- *ERLARS 2013 - 6th International Workshop on Evolutionary and Reinforcement Learning for Autonomous Robot Systems*  
Nils T. Siebel and Yohannes Kassahun
- *Fundamentals of Collective Adaptive Systems*  
Emma Hart and Ben Paechter
- *HSB - 2nd International Workshop on Hybrid Systems and Biology*  
Thao Dang and Carla Piazza
- *Protocells: Back to the Future*  
Timoteo Carletti, Alessandro Filisetti, Norman Packard and Roberto Serra
- *What Synthetic Biology can offer to Artificial Intelligence? Perspectives in the Bio-Chem-ICT and other scenarios*  
Luisa Damiano, Pasquale Stano and Yutetsu Kuruma

## Program Committee

An event like ECAL'13 would not have been possible without the following dedicated members of the Program Committee. Our gratitude goes to all of them.

Hussein Abbass	Chiara Damiani	Natalio Krasnogor	Paolo Provero
Alberto Acerbi	Thomas Dandekar	Renaud Lambiotte	Palaniappan Ramaswamy
Andy Adamatzky	Christian Darabos	Doron Lancet	Vitorino Ramos
Youhei Akimoto	Kerstin Dautenhahn	Pier Luca Lanzi	Steen Rasmussen
Lee Altenberg	Joachim De Beule	Doheon Lee	Thomas S. Ray
Michele Amoretti	Francisco Fernandez de Vega	E. Stanley Lee	John Rieffel
Martyn Amos	Kalyanmoy Deb	Jonathan Lee	Laura A. Ripamonti
Claes Andersson	Jordi Delgado	Niles Lehman	Luis M. Rocha
Marco Antonietti	Ralf Der	Tom Lenaerts	Andrea Roli
Paolo Arena	Barbara Di Camillo	Lukas Lichtensteiger	Pierre Rouze
Takaya Arita	Gianni Di Caro	Pietro Liò	Kepa Ruiz Mirazo
Dirk Arnold	Cecilia Di Chio	Hod Lipson	Franck RUFFIER
Giuseppe Ascia	Ezequiel Di Mario	Dapeng Liu	Erol Sahin
Jaume Bacardit	Peter Dittrich	Joseph Lizier	Corrado Santoro
Gianluca Baldassarre	Marco Dorigo	Daniel Lobo	Francisco C. Santos
Wolfgang Banzhaf	Alan Dorin	Fernando Lobo	Hiroki Sayama
Xabier E. Barandiaran	René Doursat	Pier Luigi Luisi	Thomas Schmickl
Helio Jose Barbosa	Marc Ebner	Torbjörn Lundh	Marc Schoenauer
Jake Beal	Pascale Ehrenfreund	Dario Maggiorini	Luis Seabra Lopes
Lucia Beccai	Gusz Eiben	George D. Magoulas	Roberto Serra
Mark Bedau	Arantza Etxeberria	Vittorio Maniezzo	Hsu-Shih Shih
Randall Beer	Giovanni M. Farinella	Elena Marchiori	Andrew Shreve
Katie Bentley	Nazim Fates	Omer Markovitch	Ricard Solé
Peter Bentley	Harold Fellermann	Davide Marocco	Giandomenico Spezzano
Heder S. Bernardino	Chrisantha Fernando	Dominique Martinez	Antoine Spicher
Daniel Berrar	Paola Festa	Antonio Masegosa	Peter Stadler
Hugues Bersini	Grazziela Figueredo	Jerzy Maselko	Kenneth Stanley
Luc Berthouze	Alessandro Filisetti	Sarah Maurer	Pasquale Stano
Mauro Birattari	Christoph Flamm	Giancarlo Mauri	Luc Steels
Jacek Blazewicz	Francesco Fontanella	John McCaskill	Giovanni Stracquadanio
Leonidas Bleris	Enrico Formenti	Chris McEwan	Reiji Suzuki
Christian Blum	Luigi Fortuna	Barry McMullin	El-Ghazali Talbi
Joshua C. Bongard	Giuditta Franco	Peter William McOwan	Arvydas Tamulis
Terry Bossomaier	Walter Frisch	José F. Mendes	Kay Chen Tan
Paul Bourgin	Ruedi Fuchsli	Olivier Michel	Uwe Tangen
Anthony Brabazon	Toshio Fukuda	Martin Middendorf	Charles Taylor
Andrea Bracciali	Luca Gambardella	Orazio Miglino	Tim Taylor
Juergen Branke	Jean-Gabriel Ganascia	Julian Miller	Pietro Terna
Larry Bull	Nicholas Geard	Marco Mirolli	German Terrazas Angulo
Seth Bullock	Philip Gerlee	Natasa Miskov-Zivanov	Christof Teuscher
Tadeusz Burczyński	Carlos Gershenson	Francesco Mondada	Gianna M. Toffolo
Stefano Cagnoni	Mario Giacobini	Luis Moniz Pereira	Marco Tomassini
Yizhi Cai	Onofrio Gigliotta	Sara Montagna	Vito Trianni
Raffaele Calabretta	Alex Graudenzi	Jason H. Moore	Soichiro Tsuda
Alexandre Campo	Roderich Gross	Giovanni Muscato	Elio Tuci
Angelo Cangelosi	Thilo Gross	Iñaki Navarro	Ali Emre Turgut
Giulio Caravagna	Mario Guarracino	Chrystopher L. Nehaniv	Karl Tuyls
Timoteo Carletti	Alaa Abi-Hadar	Giuseppe Nicosia	Jon Umerez
Alberto Castellini	Jin kao Hao	Martin Nilsson Jacobi	Renato Umeton
Vincenzo Catania	Inman Harvey	Jason Noble	Ashish Umre
Ciro Cattuto	Paulien Hogeweg	Stefano Nolfi	Edgar E Vallejo
Uday Chakraborty	Gregory S. Hornby	Wieslaw Nowak	Sergi Valverde
Bernard Chazelle	Phil Husbands	Michael O'Neill	Patricia A. Vargas
Antonio Chella	Tim J. Hutton	Ekkehard Olbrich	Richard Vaughan
Ying-ping Chen	Fumiya Iida	Ping-Feng Pai	Marco Villani
Tang-Kay Chen	Hiro Iizuka	Wei Pang	Mirko Viroli
Tianshi Chen	Takashi Ikegami	Elisa Pappalardo	Paul Vogt
Gregory S. Chirikjian	Christian Jacob	Luca Patanè	Richard Watson
Sung-Bae Cho	Yaochu Jin	Marco Pavone	Janet Wiles
Anders L. Christensen	Colin Johnson	Mario Pavone	Alan Winfield
Dominique Chu	Laetitia Jourdan	Joshua Payne	Rachel Wood
David Merodio Codinachs	Janusz Kacprzyk	David Pelta	Andrew Wuensche
David Cornforth	George Kampis	Andrew Philippides	Larry Yaeger
Luis Correia	Istvan Karsai	Simone Pigolotti	Hector Zenil
Jole Costanza	Jozef Kelemen	Raphaël Plasson	Tom Ziemke
Vincenzo Cutello	Serge Kernbach	Alessio Plebe	
Alberto D'Onofrio	Didier Keymeulen	Daniel Polani	
Bruce Damer	DaeEun Kim	Mikhail Prokopenko	

## Organizing Committee

### Chairs

- Pietro Liò, *University of Cambridge, UK*
- Orazio Miglino, *University of Naples “Federico II”, Italy*
- Giuseppe Nicosia, *University of Catania, Italy*
- Stefano Nolfi, *ICST-CNR, Italy*
- Mario Pavone, *University of Catania, Italy*

### Workshop Chair

- Giovanni Stracquadanio, *Johns Hopkins University, USA*

### Tutorial Chair

- Giuseppe Narzisi, *Cold Spring Harbor Laboratory, USA*

### Local Organizing Committee

- Claudio Angione, *University of Cambridge, UK*
- Giovanni Carapezza, *University of Catania, Italy*
- Piero Consoli, *University of Catania, Italy*
- Jole Costanza, *University of Catania, Italy*
- Marisa Lappano Anile, *Associazione Angelo Marcello Anile*
- Annalisa Occhipinti, *University of Catania, Italy*

### Publicity Chair

- Giovanni Murabito, *Di.Gi. Apps Inc.*

ECAL 2013 General Chairs:

Pietro Liò, Orazio Miglino, Giuseppe Nicosia,  
Stefano Nolfi, Mario Pavone,  
*Taormina, September 2013.*

# Cooperation and the Division of Labour

Simon J. Tudge<sup>1</sup>, Richard A. Watson<sup>1</sup> and Markus Brede<sup>1</sup>

<sup>1</sup> The University of Southampton, Southampton SO17 1BJ, UK  
sjt4g11@soton.ac.uk

## Abstract

Cooperation is vital for maintaining the integrity of complex life forms. In many cases in nature cooperation manifests itself through constituent parts performing different, but complementary, functions. The vast majority of studies on the evolution of cooperation, however, look only at the special case in which cooperation manifests itself via the constituent parts performing identical tasks. In this paper we investigate a class of games in which the socially optimal behaviour has the property of being heterogeneous. We show that this class of games is equivalent to a region of ST space (the space of normalised two-player games characterised by the ‘sucker’ and ‘temptation’ payoffs) which has previously been dismissed. We analyse, through a simple group selection model, properties that evolving agents would need to have in order to “solve” this dilemma. Specifically we find that positive assortment on pure strategies may lower mean individual payoff, and that assortment on mixed strategies will increase payoff, but not maximise it.

## Introduction

Division Of Labour (DOL) is ubiquitous in the biological world. Social insects often have specialised castes for performing individual tasks (Hölldobler and Wilson, 2009). Multicellular organisms exhibit high levels of cell differentiation. Colonial marine invertebrates have differentiated parts which also specialise (Dunn and Wagner, 2006). Even bacteria have been shown to exhibit specialisation (Crespi, 2001). Arguably DOL is one of the major benefits to group living. It has long been recognised that specialisation may result in gains in efficiency; the idea can be traced at least as far back as Adam Smith’s *Wealth of Nations* (Smith, 1776). However, with all group living comes the potential for the emergence of cooperative dilemmas. Whenever a task is broken down into smaller parts the products of the sub-tasks must be shared or distributed. This potentially opens the door to free riders who benefit from the distribution of the products of labour, without contributing to its costs.

There are a growing body of artificial life studies concerning the evolution of the division of labour. Specifically authors have addressed: the mechanisms by which a division of labour can occur (Goldsby et al., 2012), the evolu-

tionary pathway to the emergence of complex internal features (Lenski et al., 2003), the evolution of differentiation in multicellular organisms (Ray and Hart, 1999), the role of gene networks in multicellular development (Joachimczak and Wróbel, 2008) and the evolutionary role of asymmetric cell division (Hotz, 2004).

DOL is also one of the key theoretical ideas behind the major evolutionary transitions research program (Maynard Smith and Szathmari, 1997). A major transition is one in which biological entities which were, preceding the transition, able to replicate as individuals are, after the transition, only able to replicate as part of a larger whole. DOL is likely to be one of the key concepts that leads to a deeper understanding of the major transitions. As increased specialisation develops, individuals become increasingly dependent upon one another, to the point where it is no longer sensible to regard them as functionally independent entities. For example, a potentially defining characteristic of certain types of major transition (i.e. the fraternal transitions (Queller, 1997)) is a reproductive division of labour (Michod, 2006).

Cooperative dilemmas are the class of games in which well-mixed populations of agents evolve to a state which does not maximise mean individual payoff. Theoretical considerations regarding the evolution of cooperation posit a game in which the socially optimal behaviour, for the population, is for every agent to perform the action labelled as cooperate. These set of games are cooperative dilemmas if there exists an ESS which is different from total cooperation. That is that under freely evolving conditions the population is composed either partially or entirely of defectors. Models of the evolution of cooperation then typically consider extensions of the underlying game which result in an increase in the level of cooperation. A common way in which this is achieved is through imposing population structure which leads to positive assortment and hence to an increase in cooperation (see for instance: Nowak and May (1992); Maynard Smith (1964)). Here positive assortment means that like strategies play each other more often than would be expected from random interactions. (For general arguments concerning the role of assortment in the evolution

of cooperation see: Eshel and Cavalli-Sforza (1983); Queller (1985); Dugatkin and Mesterton-Gibbons (1992); Godfrey-Smith (2008)).

Despite the large variety of computational and analytic models along these lines all such studies are built on a common assumption: namely the final optimal state is a homogeneous one in which all individuals play the same strategy. However, many situations in nature can be said to be in cooperative states, but between individuals or components which are not exhibiting homogeneous behaviours. To the best of the authors' knowledge none of these evolutionary game-theoretic investigations have considered situations in which a heterogeneous final state is desirable.

This paper firstly identifies the class of games in which a mixed state is socially optimal, i.e. games in which a division of labour may evolve. We show that these games are related to the conventional cooperative dilemmas. We then go on to present two models to illustrate some key points. The first model challenges the assumption that positive assortment on pure strategies will lead to an increase in the population's mean payoff. The second model extends this by introducing the additional assumption of mixed strategies. The model shows that positive assortment on mixed strategies does lead to an increase in the population's mean payoff. Finally we sketch some further theoretical considerations which show that, although positive assortment on mixed strategies does lead to an increase in payoff, it is not the highest payoff that can be reached under any circumstance. Specifically in order to maximise average payoff individuals would have to control not just the frequency of strategies, but the frequency of interactions within the population. In this case a negative assortment on social strategies is optimal for the population; however, we show that it is not evolutionarily stable. In order for this optimal configuration to be stable it is necessary to have a higher level positive assortment on genotypes which provides a lower level negative assortment of phenotype/social strategy.

### Division of Labour Games

We now outline a formalism which enables us to think about the division of labour in the simplest non-trivial case.

Consider a situation in which individuals meet and perform one of two tasks: A or B. Each task bestows a benefit to both of the individuals involved in the interaction. The benefits are given by  $b_A$  and  $b_B$  respectively. Each individual must bear the cost of their performed task themselves. Costs are given by  $c_A$  and  $c_B$ . However, if both individuals perform the same task the cost of that task is shared between them. In addition, there is a synergistic benefit which is the benefit of having both tasks performed together:  $\delta$ . We consider the cases in which A has a higher cost but also a higher benefit than B, i.e.  $c_A > c_B$  and  $b_A > b_B$ .

An example of the situation described above might go as follows. Two human individuals living in the same tribe may

perform one of two tasks. Task A is to go and hunt for meat. Task B is to build a fire. Hunting comes at an extra cost to the individual either because it requires more energy or because it is inherently riskier. The benefit of hunting is meat. Building the fire has a lower, but non-zero, cost. The benefit of building the fire is warmth. We assume that the meat is more valuable than the warmth, but that both tasks provide some benefit in isolation. In this instance the synergistic benefit,  $\delta$ , is that of having cooked meat. It is the benefit above and beyond that of the sum of the two benefits in isolation. We are assuming here that the benefits are non-excludable, that is that the hunter could not stop the fire builder from taking meat, and vice versa. This cartoon is an aid to understanding; the essential features of the situation are represented via the payoff matrix:

	A	B
A	$b_A - c_A/2$	$b_A + b_B + \delta - c_A$
B	$b_A + b_B + \delta - c_B$	$b_B - c_B/2$

Given that games have two arbitrary degrees of freedom we will assume that:  $b_A - c_A/2 = 1$  and  $b_B - c_B/2 = 0$ . We can thus rewrite the above payoff matrix as:

	A	B
A	1	$1 - r + \delta$
B	$1 + r + \delta$	0

Where  $r = \frac{1}{2}(c_A - c_B)$ . This reduces to the 1 dimensional parameterisation of the snowdrift game if  $\delta = 0$  (Hauert, 2004). Conceptually we may also think of the story behind the snowdrift game as the special case in which task B is the task of doing nothing with no benefit and no cost.

We can trivially see that this game represents a region of ST space using  $S = 1 - r + \delta$  and  $T = 1 + r + \delta$ . For an explanation of ST space see Santos et al. (2006).

Note that  $r$  is the difference in cost between performing the two tasks, and can thus be thought of as parameterising the severity of the dilemma.  $\delta$  represents the synergistic benefits of having both tasks performed.  $\delta > 0$  corresponds to the region  $S + T > 2$ .



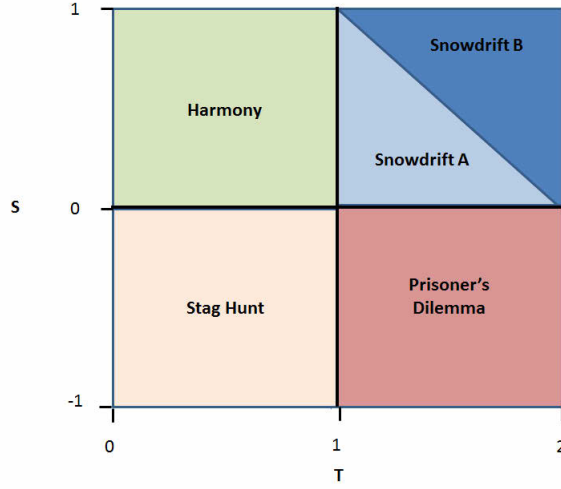


Figure 1: The location of all games in ST space. Previously the top right quadrant was simply referred to as the snowdrift game. Here we split this region into two. Snowdrift A corresponds to games which are snowdrift games but not DOL games. Snowdrift B refers to snowdrift games which are also division of labour games. The remainder of this paper is concerned with the region labelled snowdrift B.

### Cooperative Dilemmas

The previous section introduced a class of games which are formally equivalent to the region of ST space in which  $S + T > 2$ . We shall refer to these games as Division of Labour games. Snowdrift games are defined by  $0 < S < 1$  and  $T > 1$ . We shall focus our investigation on the class of games which are both snowdrift games and DOL games (note that neither one implies the other).

ST space was conceived of in order to systematically investigate all classes of cooperative dilemmas. Some authors (see for instance Macy and Flache (2002)) specifically exclude the region  $S + T > 2$  from the definition of cooperative dilemmas. We find this exclusion somewhat artificial. The essence of a cooperative dilemma is a situation in which evolution leads to a state which does not maximise mean individual payoff. It just so happens that in conventional cooperative dilemmas social welfare is maximised by every agent cooperating, but this is by no means an essential part of the argument. Let us define the Socially Optimal Frequency (SOF) as the frequency of cooperate (or type A in the language of DOL games) which maximises the mean payoff of the population under well mixed conditions. We then define a cooperative dilemma as one in which  $\text{SOF} \neq \text{ESS}$ .

We now need to derive an equation for the SOF in terms of  $S$  and  $T$ . To do this note that the mean fitness of the population is given by:

$$\bar{f} = f_c \rho_c + f_d \rho_d \quad (1)$$

We consider a fixed population size so that  $\rho_d = 1 - \rho_c$ . Fitnesses are given by  $f_c = \rho_c + S\rho_d$  and  $f_d = T\rho_c$ . We then arrive at a formula for average fitness in terms of density of cooperators ( $\rho$ ):

$$\bar{f} = \rho(S + T + \rho(1 - S - T)) \quad (2)$$

The SOF is the maximum of this function for  $\rho \in [0, 1]$ . It is straight forward to prove that:

$$\text{SOF} = \begin{cases} \frac{S+T}{2(S+T-1)} & S+T > 2 \\ 1 & S+T \leq 2 \end{cases} \quad (3)$$

The ESS of the snowdrift game is (Nowak, 2006a):

$$\text{ESS} = \frac{S}{S+T-1} \quad (4)$$

for DOL games which are also snowdrift games the ESS is only equal to the SOF in the very special instance in which  $S = T$ . Thus, by our slightly broader definition, DOL games are cooperative dilemmas.

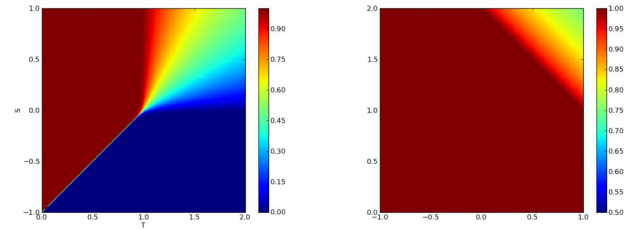


Figure 2: Left: The equilibrium frequency of cooperate. Right: the SOF, note that in the top right hand corner this is not equal to 100% cooperation.

### Model

In this section we demonstrate that positive assortment on pure strategies is only sufficient in allowing populations to reach the SOF in the non-generic case in which  $\text{SOF} = 1$ . We go on to show that positive assortment is still one of the key elements in allowing populations to reach the SOF. However, in this case the assortment must be on something other than pure social strategies.

We implement two generational GAs (labelled model I and II) to illustrate some key points. In both cases we construct a scenario in which one can exogenously control the level of assortment in a population of evolving individuals and measure the total payoff in the population. Model I serves as a control for model II. With model I we allow only pure strategies which are only able to perform one of two tasks, A or B, for the entirety of their lifetime. In model II we lift this assumption and allow for mixed strategies. Specifically a genotype specifies not a task A or B, but a probability,  $p \in [0, 1]$ , which determines how often task A is

performed. Apart from this difference models I and II have the same underlying structure.

The models consist of two distinct phases, a group phase and a population phase. Rounds of the game during one generation occur in  $\tau$  discrete time steps; the first  $X$  time steps within groups, the remaining  $\tau - X$  time steps within the fully mixed population. If  $f_G$  is fitness acquired within the group phase and  $f_P$  fitness acquired within the population phase then total fitness,  $f_T$ , is given by:

$$f_T = \frac{X}{\tau} f_G + \frac{\tau - X}{\tau} f_P \quad (5)$$

There are  $N_G$  groups consisting of  $g$  players in each group. Individuals acquire fitness over both stages of the generation.  $N_G$  individuals are chosen at the end of the population stage via fitness proportionate selection and go on to form new groups. The founding individuals immediately replicate  $g - 1$  times so that the groups are composed of  $g$  clonal individuals. Figure 3 shows schematics for the two models.

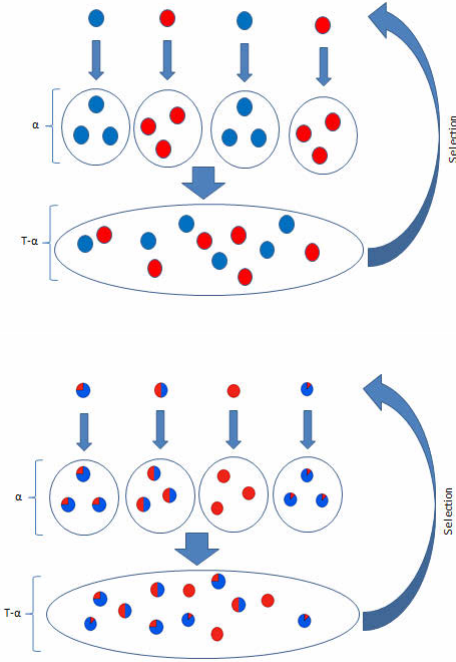


Figure 3: Schematics for models I and II respectively.

We consider the model in two different manners. First of all, we model the situation through numerical integration of the relevant replicator equation (see appendix A). This corresponds to infinite populations without mutations. Secondly, we model the system via an agent based simulation with finite population and mutations. The two approaches show good agreement in final results.

In the agent-based model there is mutation. Mutation

leads to one of the individuals in the group stage being genetically different to their parent.

In model I we model mutation by allowing an A to create a B, and vice versa, with a probability  $\mu = 1 \times 10^{-2}$ . In model II, with probability  $\mu = 5 \times 10^{-2}$ , an individual is born with a value of  $p$  which differs from its parent by an amount chosen from the random uniform distribution  $[-0.1, 0.1]$ . Mutations are capped to physically meaningful ranges (i.e. between 0 and 1) if they mutate outside of this range.

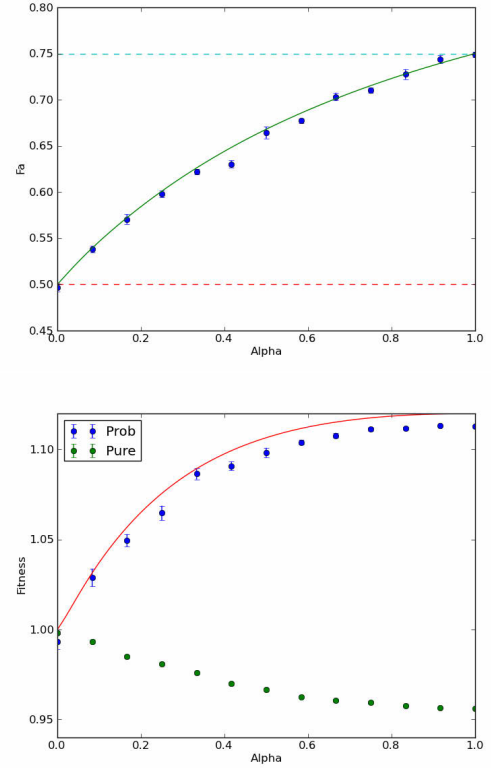


Figure 4: Top: The mean value of  $p$  at equilibrium for a range of values of  $\alpha$  in model II. The lower dotted line is the ESS and the upper one the SOF. Bottom: Fitness at ESS for the two models. The solid line is generated from the predictions of the analytic model, points from the agent based model. Clearly agents in model II are at an advantage over those in model I.  $(r, \delta) = (0.5, 0.5)$ .

This group structuring model allows us to fine-tune the level of assortment on the population. It bears some similarities to the hay-stack model (Maynard Smith, 1964). However, within a group there is no selection, as all members are clonal. The groups are formed from a founder and serve only to limit the interactions of individuals to a certain, non-random, subset of the population. In this sense the models bear some conceptual similarities to the ones discussed in Godfrey-Smith (2008). The qualitative results would be repeatable with any of the standard repertoire of “evolution of

cooperation” models (Nowak, 2006b). This particular model is not chosen for biological realism, but because the essential property of population structure leading to positive assortment is completely transparent. The models have the convenient feature of being able to tune the level of population structure via the parameter  $\alpha = x/\tau$ .

Previously the distinction between pure and mixed strategies has not been important. In a snowdrift game the frequency of cooperate at the ESS can represent either the frequency of pure strategy cooperators within the population or the average value of  $p$  (the probability of cooperating) within a population of mixed strategy individuals (Maynard Smith, 1982, p.17). These two models show that this distinction is important when playing a game in which  $\text{SOF} \neq 1$ .

In model I there is a one-to-one correspondence between the genotype and the social strategies A and B. Group structure provides only positive assortment on pure strategies. Assortment is needed for the evolution of cooperation. However, positive assortment leads to groups comprised only of one type.

It is important to realise that model II provides assortment on mixed strategies rather than on the pure phenotypes A and B. This is crucial to the following results.

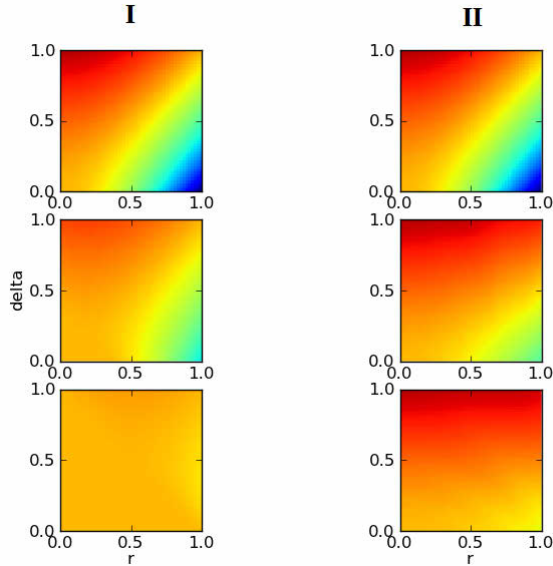


Figure 5: The mean fitness at ESS. Left column is for pure strategies, as in model I, right column is for mixed strategies, as in model II. Going down the page the figures correspond to increasing levels of  $\alpha$ . Top:  $\alpha = 0$ , middle:  $\alpha = 0.25$  and bottom:  $\alpha = 0.5$ .

Figure 4 shows the results of the two versions of the model. There are 125 groups composed of 5 individuals each. The simulation is run for  $5 \times 10^6$  generations and the

average value of  $p$  (probability of playing strategy A) over the entire population is recorded. We find that groups in model II evolve towards the SOF for larger values of  $\alpha$  (the degree of population structure). As expected with  $\alpha = 0$  the population simply evolves to the ESS in both models.

We plot the relative fitness for models I and II for increasing  $\alpha$  to illustrate the fact that mixed strategies are at an evolutionary advantage over pure ones. Note that by construction pure strategies only interact with types of the same pure strategy within groups (in model I). In the case where the socially optimal solution was pure cooperate the ability to form pure groups is sufficient to solve the dilemma. However, in general these are a special type of game. For DOL games, in which a mix of strategies is desirable, mixed strategies can outperform pure ones.

Figure 5 shows the fitness at ESS for models I and II for all games parameterised via  $r$  and  $\delta$ . Interestingly positive assortment on pure strategies can actually be detrimental to the population’s payoff if  $\delta$  (the synergistic benefit to heterogeneous behaviours) is sufficiently high. The higher the synergistic benefit is the greater the advantage of having mixed strategies.

## Group Phenotypes

Groups composed of mixed strategies do not maximise social welfare. The reason for this is that they are unable to control their internal structure or organisation. In this section we formalise this point. We leave a detailed specification and analysis of a model for a forthcoming work.

In a well-mixed population the parameter  $\rho$  (frequency of type A) characterises the state of the system. However, if interactions are not random then  $\rho$  on its own is insufficient. We also need to know the frequency of the different types of within-group interactions. In principle there are three types of interactions: (A-A, A-B and B-B). If the total number of interaction of all types is fixed, then knowing the fractional density of each type will specify the state of the group. We will denote these three densities as  $\phi_{AA}$ ,  $\phi_{AB}$  and  $\phi_{BB}$ . However, it is sufficient to know only one of these. Let us then use  $\phi_{AB}$  and drop the subscript. We shall define the *group phenotype* as a point in the space  $(\rho, \phi)$ . The following formulae show how the densities of all types of interaction can be found from these two variables.

$$\phi_{AA} = \rho - \frac{1}{2}\phi \quad (6)$$

$$\phi_{AB} = \phi \quad (7)$$

$$\phi_{BB} = 1 - \rho - \frac{1}{2}\phi \quad (8)$$

Notice that  $\phi$  is confined within certain ranges based on  $\rho$ . Specifically  $0 \leq \phi \leq 2\text{Min}\{\rho, 1 - \rho\}$ .  $\phi$  is equivalent to certain measures of linkage disequilibrium (see for instance Hartl and Clark (1998)). For interesting parallels between population genetics and the evolution of cooperation

see Gardner et al. (2007).

The previous section considered the cases in which groups only had control over the parameter  $\rho$  and had no way of controlling the internal composition of within group interactions. In this case they were confined to have a value of

$$\varphi = \varphi^{(R)} = 2\rho(1 - \rho) \quad (9)$$

where R stands for random. Thus the SOF (Socially Optimal Frequency) corresponds to the fittest group with random internal interactions. Let us also define the Optimal Group Phenotype (OGP) to be the point in  $(\rho, \varphi)$  space which maximises average fitness. To see this, note that average fitness is given by:

$$\bar{f} = \rho + \frac{1}{2}(S + T - 1)\varphi \quad (10)$$

for games in  $(S, T)$  space. The OGP is then given by:

$$OGP = \begin{cases} (1, 0) & S + T < 2 \\ (1/2, 1) & S + T > 2 \end{cases} \quad (11)$$

thus in DOL games a group which maximises the amount of A-B links at the expense of all other types of interactions is the one which maximises group fitness.

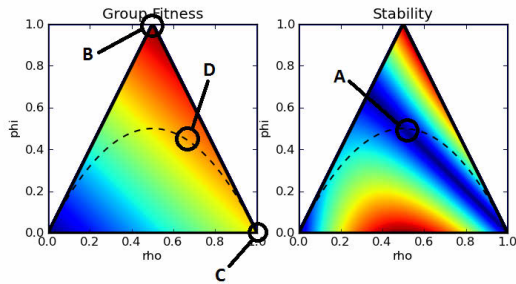


Figure 6: Left: The average fitness in terms of  $\rho$  and  $\varphi$ , for a certain game in which  $R + T > 2$ . Right: the absolute difference between the fitness of the two phenotypes A and B (assuming that the group is composed of pure strategy individuals). The dotted line marks  $\varphi^{(R)}$ . Circled points correspond to A: The ESS under well mixed conditions, B: The SOF, C: The optimal point for pure groups and D: the OGP. Note: if  $S + T < 2$  B, C and D coincide. The fact that stability is non-zero at points B and D shows that these situations are fundamentally unstable without the addition of extra assumptions.

With reference to figure 6 model I of the previous section was only able to evolve to point C, model II was able to do

better and reach point B. However, neither model produced groups who were able to reach point D. This is the subject of a forthcoming work in which we investigate the effects of developmental or aggregational processes.

## Discussion

Kant said:

“Act only according to that maxim whereby you can, at the same time, will that it should become a universal law.”

In a fully assorted population Kant’s principle is not only morally commendable, but it is also entirely sensible. Given that you will only meet individuals who are the same as you it makes sense to perform social actions which are beneficial to be on the receiving end of. Thus it would seem that positive assortment is *the* answer to the evolution of cooperation.

On the other hand gains from specialisation occur via a collection of *different* types of individuals. We have seen that a division of labour game may be a cooperative dilemma. There are two needs which seem to be fundamentally at odds with each other: firstly, the need for positive assortment to alleviate the cooperative dilemma, and secondly, the need for negative assortment in order to gain from specialisation. This is the fundamental problem of the evolution of the division of labour. How does nature have her cake and eat it? That is how does evolution create the positive assortment necessary to alleviate the cooperative dilemma, but at the same time maintain the diversity needed to benefit from a division of labour?

It could be argued that many interesting and complex aspects of the biological world are about solving this problem. Phenotypic plasticity is a way in which a social strategy is able to become decoupled from the genotype which underlies it (Gavrilets, 2010). Thus we can have assortment on genotype without assortment on phenotype (as in model II), which goes some way to alleviating the problem of the division of labour. This is the key point which the models presented here attempt to illustrate. One way of expressing this would be to say that the social strategy has become de-Darwinised (*sensu* Godfrey-Smith (2009)). In division of labour games the optimal configuration involves As interacting with Bs to the exclusion of all other types of interaction. However, in this situation the fitness acquired by the phenotype B will always outweigh that acquired by A (because in our framework  $T > S$ ). Thus the optimisation of the higher level entity, the group, is in direct conflict with that of the lower level entities, the individuals. The only way in which higher level optimisation can occur is if selection does not act directly on the frequency of the constituent types A and B. This is what we mean by de-Darwinisation. A potential way for this to occur is through a genotype-phenotype map which is not one-to-one (i.e. a genotype may specify more



than one phenotype). In this case, although social interactions lead to Bs having a higher fitness, the configuration is sustainable because natural selection does not “see” the phenotypes A and B, it only “sees” genotypes which specify certain frequencies and organisations of social strategies. We see the concept of de-Darwinisation as a powerful conceptual tool for understanding the emergence of higher levels of biological organisation.

An ALife approach will doubtless be one of the key theoretical tools in our quest to understand biological organisation. Simulation is necessary not only because the processes of interest are obscured by time, but also because we only have one truly independent example of life. What we really want to know is which aspects of biology are contingent on the particulars of our bio-chemistry, and which are profound consequences of the logic of natural selection. This paper has attempted to add to the small, but growing number, of ALife studies which tackle the question of the division of labour and internal organisation. We have laid down ground work for a systematic investigation of the ultimate causes of the evolution of internal differentiation and organisation.

### Acknowledgments

This work was supported by an EPSRC Doctoral Training Centre grant (EP/G03690X/1). Thanks to Paul Ryan and Patrick Doncaster for their useful comments. Also to Charlott Rittmeister for her proof reading.

### Appendix A

We sketch the solution to the model via the replicator equation formalism, for the case of mixed strategies playing within variable levels of group structure.

Individuals have strategies which are specified via a probability  $p$ . They play strategy A with probability  $p$ , and therefore B with probability  $1 - p$ . An individual with strategy  $p$  who interacts with another individual with strategy  $q$  receives an expected payoff of:

$$F[p, q] = pqR + p(1 - q)S + (1 - p)qT + (1 - p)(1 - q)P \quad (12)$$

Selection acts on  $p$  and thus the population is specified by the function  $\rho(p)$  which is a 1D function which specifies the density of the population playing a strategy for every value of  $p \in [0, 1]$ . The systems dynamics are specified by the replicator equation:

$$\dot{\rho}(p) = \rho(p) (f(p) - \bar{f}(p)) \quad (13)$$

where  $f(p)$  is the fitness of the individuals for a given  $p$ , and  $\bar{f}(p)$  is the mean fitness of the population.

The fitness of any strategy comprises of two parts. The fitness gained in the group phase, and the fitness gained in the population phase. Call these fitness  $f_G$  and  $f_P$  respectively. In the absence of mutations strategies always play with like

strategies in the group phase, thus:

$$f_G = F[p, p] \quad (14)$$

in the population phase strategies play with every other strategy. The average payoff is given by the strategy they would have received from playing a hypothetical average individual. That is:

$$f_P = F[p, \bar{p}] \quad (15)$$

where  $\bar{p}$  is the average value of  $p$  in the population.

Total fitness is thus:

$$f(p) = T_G F[p, p] + T_P F[p, \bar{p}] \quad (16)$$

we normalise by saying that  $T_G + T_P = \tau$ , i.e. that the whole cycle happens over  $\tau$  units of time. Let  $\alpha = T_G/\tau$ . By dividing by  $\tau$  we arrive at:

$$f(p) = \alpha F[p, p] + (1 - \alpha) F[p, \bar{p}] \quad (17)$$

We thus have a fitness defined for every possible strategy, which can be used to model the situation by means of the replicator equation.

Population dynamics follow from numerical integration of equation 13.

### References

- Crespi, B. J. (2001). The evolution of social behavior in microorganisms. *Trends in ecology & evolution*, 16(4):178–183.
- Dugatkin, L. A. and Mesterton-Gibbons, M. (1992). Cooperation Among Unrelated Individuals: Evolutionary Factors. *The Quarterly Review of Biology*, 67(3):267–281.
- Dunn, C. W. and Wagner, G. P. (2006). The evolution of colony-level development in the Siphonophora (Cnidaria:Hydrozoa). *Development genes and evolution*, 216(12):743–54.
- Eshel, I. and Cavalli-Sforza, L. L. (1983). Assortment of encounters and evolution of cooperativeness. *Proceedings of the National Academy of Sciences*, 79(4):1331–1335.
- Gardner, A., West, S. A., and Barton, N. H. (2007). The relation between multilocus population genetics and social evolution theory. *The American naturalist*, 169(2):207–26.
- Gavrilets, S. (2010). Rapid transition towards the Division of Labor via evolution of developmental plasticity. *PLoS computational biology*, 6(6).
- Godfrey-Smith, P. (2008). Varieties of Population Structure and the Levels of Selection. *The British Journal for the Philosophy of Science*, 59(1):25–50.

- Godfrey-Smith, P. (2009). *Darwinian Populations and Natural Selection*. OUP Oxford.
- Goldsby, H. J., Dornhaus, A., Kerr, B., and Ofria, C. (2012). Task-switching costs promote the evolution of division of labor and shifts in individuality. *Proceedings of the National Academy of Sciences of the United States of America*.
- Hartl, D. L. and Clark, A. G. (1998). *Principles of Population Genetics*. Sinauer Associates.
- Hauert, C. (2004). Spatial structure often inhibits the evolution of cooperation in the snowdrift game. *Nature*.
- Hölldobler, B. and Wilson, E. (2009). *The Super-Organism*. Norton.
- Hotz, P. (2004). Asymmetric cell division in artificial evolution. *Evolutionary Computation, 2004. CEC2004. . . .*
- Joachimczak, M. and Wróbel, B. (2008). Evo-devo in silico: a model of a gene network regulating multicellular development in 3D space with artificial physics. *Artificial Life XI: Proceedings of the . . .*, pages 297–304.
- Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423(6936):139–44.
- Macy, M. W. and Flache, A. (2002). Learning dynamics in social dilemmas. *Proceedings of the National Academy of Sciences of the United States of America*, 99 Suppl 3:7229–36.
- Maynard Smith, J. (1964). Group selection and kin selection. *Nature*.
- Maynard Smith, J. (1982). *Evolution and the Theory of Games*. Cambridge University Press, Cambridge, UK.
- Maynard Smith, J. and Szathmary, E. (1997). *The Major Transitions in Evolution*. OUP Oxford.
- Michod, R. E. (2006). The group covariance effect and fitness trade-offs during evolutionary transitions in individuality. *Proceedings of the National Academy of Sciences of the United States of America*, 103(24):9113–7.
- Nowak, M. A. (2006a). *Evolutionary Dynamics: Exploring the Equations of Life*. Belknap Press.
- Nowak, M. A. (2006b). Five rules for the evolution of cooperation. *Science (New York, N.Y.)*, 314(5805):1560–3.
- Nowak, M. A. and May, R. (1992). Evolutionary games and spatial chaos. *Nature*.
- Queller, D. C. (1985). Kinship, reciprocity and synergism in the evolution of social behaviour. *Nature*, 318.
- Queller, D. C. (1997). Cooperators since life began. *Quarterly Review of Biology*.
- Ray, T. and Hart, J. (1999). Evolution of differentiated multi-threaded digital organisms. *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems.*, 1:1–10.
- Santos, F. C., Pacheco, J. M., and Lenaerts, T. (2006). Cooperation prevails when individuals adjust their social ties. *PLoS computational biology*, 2(10):e140.
- Smith, A. (1776). *Wealth of Nations*. Hayes Barton Press.

## Autonomous construction of synthetic cell membrane

Yutetsu Kuruma<sup>1</sup>, Hideaki Matsubayashi<sup>1</sup> and Takuya Ueda<sup>1</sup>

<sup>1</sup> Department of Medical Genome Sciences, Graduate School of Frontier Sciences, The University of Tokyo, Japan  
kuruma@k.u-tokyo.ac.jp

### Abstract

A minimal artificial living cell is a sustainable and reproducible cell-like entity composed of biological components such as proteins, DNA, RNA and phospholipids (Luisi et al. (2006)). The most realistic strategy in producing such an artificial cell is assembling biomolecules that imitate the architecture and the function of biosystems in living organisms (Oberholzer et al. (1995)). Firstly we reconstructed the gene expression machinery with the minimal number of purified translation factors addressing the need for an artificial gene expression system. The PURE (Protein synthesis Using Recombinant Elements) system (Shimizu et al. (2001)), a key tool for bottom-up synthetic biology, enables information encoded in the DNA sequence to be converted to functional proteins and enzymes, and can be used in developing artificial cellular components. Another important and indispensable feature of artificial cells is the encapsulation of genetic information and gene expression system by a lipid bilayer membrane (Ishikawa et al. (2004); Kuruma et al. (2009)). This is also important to sustain an individual from environment. In addition to compartmentalization of biomolecule components, the membrane provides a structural platform for important biological functions such as selective transport of materials, adoption of environment information, production of energy, etc. Actually, many of the vital cell functions reside on the lipid membrane, and these functions mostly rely on membrane proteins.

In this paper, we focused on three important membrane functions, i.e. (i) lipid synthesis, (ii) energy production and (iii) membrane protein synthesis. All these membrane functions were indispensable for sustaining cell alive and must be reconstructed as a consequence of internal metabolic reactions. Each membrane function has the corresponding membrane proteins. Therefore our strategy is to construct the membrane function on artificial membrane vesicles, liposomes, through the gene expression of the corresponding protein components by the PURE system. (i) For the lipid synthesis, we selected eight membrane enzymes involving in the biosynthesis process of major phospholipids from a bacterial genome. These eight enzymes were synthesized by the PURE system in the presence of membrane fraction (membrane Nano-Disc) and measured their activities. The goal of this project is to develop a biochemical network *in vitro* that aims to produce several kinds of phospholipids necessary for the formation of cell envelop. As the first step, we have succeeded to synthesize two membrane enzymes inside liposomes and individually detected their activities (5). This result would be a fundamental for the construction of a self-reproducible cell mem-

brane. (ii) For energy production, a membrane embedded super molecule complex, FoF1-ATP synthase (FoF1), was synthesized through the eight kinds of component proteins. The function of FoF1 is to produce ATP molecules, which is an energy source of most cellular activities, based on the proton gradient across a membrane. We have succeeded to synthesize FoF1 complex by the PURE system and detected its ATP synthesis activity that driven by an artificially generated proton gradient. Furthermore, the reconstructed FoF1 is coupled with another membrane machinery, bacteriorhodopsin (bR), to construct an artificial organelle. The bR is proton pump machinery that transports protons to inside of the membrane vesicles due to light stimulation. Therefore, our idea is that if the bR and FoF1 were allocated on a same membrane vesicle, the resulting vesicle is able to generate ATP molecules by light irradiation (Fig. 1). In this design, we have succeeded to detect the production of ATP molecules in the rate of 35 nmol ATP/hr/mL Reaction Solution. If the produced ATP could be used for protein synthesis reaction within the PURE system, this represents an energetically independent system and becomes a practical platform of autonomous artificial cell. (iii) All these membrane machineries are built up based on a spontaneous membrane insertion of the synthesized membrane proteins. However, a certain kind of membrane protein cannot be integrated spontaneously. In that case, the membrane protein needs a help of special membrane machinery, Sec translocon, to achieve the native formation on a lipid membrane. The Sec translocon works as a gate to mediate a membrane insertion and secretion of membrane proteins (Fig. 2). Therefore our idea is to synthesize the component proteins of Sec translocon by the PURE system and construct the Sec translocon on membrane vesicles. Since most of membrane proteins are generated through Sec translocon in living cells, any types of membrane protein can be produced after the construction of Sec translocon. So far, we have succeeded to synthesize three component proteins (SecYEG) of Sec translocon of bacteria and to detect its heterotrimeric complex formation on a lipid membrane. Furthermore, so synthesized SecYEG enables to produce another membrane proteins that cannot be spontaneously integrated into the membrane. This result indicates that, by synthesizing the Sec translocon, other important membrane proteins (machineries) can be continuously produced on the artificial membrane vesicles. Using our *in vitro* gene expression system, cell membrane functions can be partially constructed on the artificial membrane vesicles. More importantly, these membrane functions were autonomously constructed just by adding of the corresponding DNAs. The ability of an artificial cell to au-

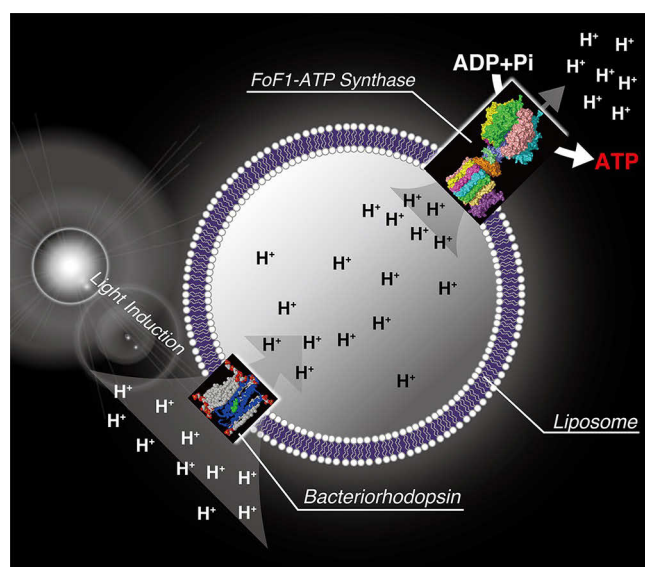


Figure 1: Artificial organelle consists of FoF1-ATP synthase and bacteriorhodopsin for production of ATP.

tonomously produce membrane protein machineries by its internal genetic/metabolic network is consistent with the theory of autopoiesis by Varela and Maturana (Varela et al. (1974)). This is adaptive and complements the definition of life, self-reproduction, which is based on gene replication. We believe that our cell-free approach will become a central device for the construction of artificial cell membranes and a breakthrough for the realization of artificial cells.

## References

- Ishikawa, K., Sato, K., Shima, Y., Urabe, I., and Yomo, T. (2004). Expression of a cascading genetic network within liposomes. *FEBS letters*, 576(3):387–390.
- Kuruma, Y., Stano, P., Ueda, T., and Luisi, P. L. (2009). A synthetic biology approach to the construction of membrane proteins in semi-synthetic minimal cells. *Biochimica et Biophysica Acta (BBA)-Biomembranes*, 1788(2):567–574.
- Luisi, P. L., Ferri, F., and Stano, P. (2006). Approaches to semi-synthetic minimal cells: a review. *Naturwissenschaften*, 93(1):1–13.
- Oberholzer, T., Albrizio, M., and Luisi, P. L. (1995). Polymerase chain reaction in liposomes. *Chemistry & biology*, 2(10):677–682.
- Shimizu, Y., Inoue, A., Tomari, Y., Suzuki, T., Yokogawa, T., Nishikawa, K., and Ueda, T. (2001). Cell-free translation reconstituted with purified components. *Nature biotechnology*, 19(8):751–755.

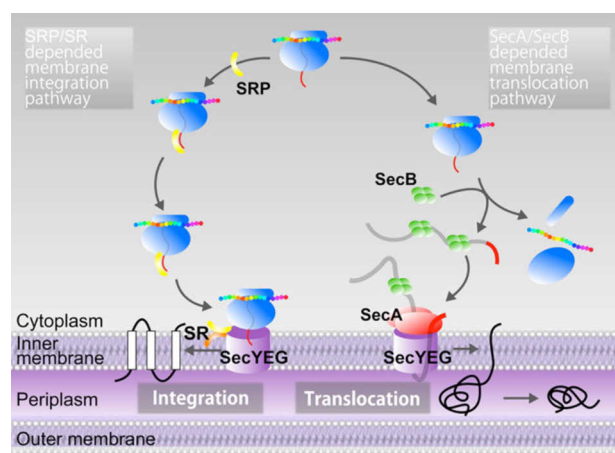


Figure 2: Membrane secretion and integration functions of SecYEG translocon.

Varela, F. G., Maturana, H. R., and Uribe, R. (1974). Autopoiesis: The organization of living systems, its characterization and a model. *Biosystems*, 5(4):187–196.



# Exploring the Role of the Tail in Bipedal Hopping through Computational Evolution

Jared M. Moore<sup>1</sup>, Anne K. Gutmann<sup>2</sup>, Craig P. McGowan<sup>2</sup>, and Philip K. McKinley<sup>1</sup>

<sup>1</sup>Dept. Computer Science and Engineering, Michigan State University, East Lansing, Michigan, USA

<sup>2</sup>Dept. Biological Sciences, University of Idaho, Moscow, Idaho, USA  
moore112@msu.edu

## Abstract

Bipedal hopping has evolved as a mode of terrestrial locomotion in relatively few mammalian species. Despite large differences in body size, habitat use, and having evolved independently, all species that use bipedal hopping have remarkably similar limb morphology and posture. In addition, these species all have relatively long tails, presumably to assist in maintaining stability. However, the evolution of this behavior, and specifically the role of the tail, is not well understood. In this paper, we explore the evolution of bipedal hopping in a simulated animat, using a relatively simple musculoskeletal model and a rigid-body physics simulation environment. Results indicate that characteristically different hopping gaits evolve with alterations to the morphology, including the structure and actuation of the tail. Many of the results are consistent with behaviors and morphologies observed in natural organisms. However, in some cases effective hopping evolved despite key differences from nature, potentially inspiring new design approaches in robotic and biomechanical systems.

## Introduction

Bipedal hopping has evolved in relatively few mammalian species, but apparently for different reasons. In small animals such as kangaroo rats (Figure 1), spring hares, and jerboas, hopping is primarily used as a predator escape mechanism (Biewener and Blickhan, 1988). In larger animals, such as kangaroos and wallabies, hopping offers an energy-efficient means of locomotion over long distances (Dawson and Taylor, 1973). Despite size differences, the overall morphologies of these animals are quite similar. Specifically, bipedal hoppers tend to have long tails and powerful hind legs, which perform the majority of work during locomotion.

Yet, the evolutionary origins of this behavior, as well as many related issues, remain obscure. Can bipedal hopping evolve only with this morphology, or is it coincidence that these various species exhibit similar body proportions? Which aspects of the morphology are essential to hopping? Do there exist other morphologies for which bipedal hopping would provide an effective means of locomotion? Not only can answering such questions inform biology, but a better understanding of the evolutionary history and mechanics

of hopping has application in biomechanics, robotics and the development of prosthetics. Unfortunately, the relatively small number of species that exhibit this behavior, as well as incomplete fossil records, make it difficult to address this problem through natural systems alone.

Computational evolution provides a means to explore the selective pressures that can lead to hopping, as well as morphological characteristics that sustain it over generations. Moreover, both the behavior and body can deviate from those occurring in nature, enabling the researcher to discover more general principles regarding these issues. A previous study into the evolution of hopping using a 2D musculoskeletal model found that both quadrupedal and bipedal hopping gaits are very sensitive to changes in morphology (Hase et al., 2004). However, such a model does not take into account many aspects of hopping, such as maintaining balance, that are essential in the physical world. Our work explores the evolution of hopping in 3D physics-based simulation environments. While our early studies, described here, rely on rigid-body physics environments, more complex musculoskeletal models have been developed (Gutmann et al., 2012) and will be integrated into our investigations as computational capacity permits.

In this paper, we focus on the role of the tail in the evolution of hopping behavior. The virtual animat model approximates muscles, joints, mass and torque, enabling us to evolve biologically plausible patterns of movement. Through a series of five evolutionary treatments, described



Figure 1: The kangaroo rat was selected as the base morphology for studying the evolution of bipedal hopping, due to its representative morphology and the availability of information on both the mechanics and dynamics of its behavior.

later, we investigate the effect of different initial (and evolvable) tail configurations on the evolution of effective hopping gaits. We initially start with a fixed morphology resembling a kangaroo rat, but restrictions on the morphology are loosened with each subsequent treatment.

The contributions of this paper are as follows. First, the proposed muscle model produces locomotion patterns similar to those of natural organisms and limits the output potential of each individual joint. This model is computationally less expensive than a musculoskeletal dynamics simulator, enabling the large number of evaluations necessary in evolutionary approaches. Second, the results demonstrate that a tail is essential to hopping, but that different configurations can lead to very different gaits, some closely resembling those of biological counterparts (namely kangaroo rats and wallabies), and others different from any known species. Third, while we observed a close coupling among tail movement and the oscillation frequency of leg joints, we discovered multiple combinations that produced effective bipedal hopping behavior. Finally, we were surprised that many evolved tails had relatively low mass, as it is hypothesized that a heavy tail helps maintain a high moment of inertia in animals, producing a more stable gait. This result might be due to our relatively simple model of the morphology (we plan to use more detailed musculoskeletal models in the future), but might also represent a combination of morphology and behavior that has application outside biology.

## Related Work

The role of the tail in locomotion is of considerable interest within biology. In their studies of geckos, which are not bipedal hoppers, Full and colleagues found that the tail is essential to both orientation control and gait stability (Jusufi et al., 2008; Libby et al., 2012). Alexander and Vernon studied the musculoskeletal system of kangaroos and described the overall mechanical system and the forces exerted during hopping (Alexander and Vernon, 1975). They also first hypothesized that the tail was necessary to balance the angular momentum produced by the swinging legs during hopping. However, to our knowledge no one has yet tested this hypothesis, nor explored its significance in other hopping species.

In robotics, hopping is an intriguing locomotion strategy for its potential energy efficiency and the ability to rapidly change elevation. The latter is particularly important to radio communication, as signal propagation distance is greatly increased by moving transmitters above ground level (Cintrón and Mutka, 2010). Indeed, research in this area has led to the development of small robots capable of both self stabilization and hopping (Zhao et al., 2009). Prior studies on hopping have also addressed mechanics of simple, single-joint actuated robots that were able to achieve stable hopping gaits (Berkemeier and Fearing, 1998), and single-hop robots have been constructed using pneumatic muscle actu-

ators (Niiyama et al., 2007). It has also been shown that combining several hops was more energy efficient than a single, powerful hop, while producing the same jumping height (Aguilar et al., 2012). This efficient hopping motion was discovered after analyzing thousands of results, lending support to harnessing the search capability of evolutionary computation in order to address similar problems.

Evolutionary approaches have been shown to be successful in many robotic and biological applications. Beginning with the foundational work of Brooks and Sims (Brooks, 1992; Sims, 1994), computational evolution has proven effective at producing a diverse range of behaviors. Examples include evolution of neural-based controllers (Cliff et al., 1993; Ijspeert, 2001, 2008) and locomotion strategies for real or simulated robots (Bongard, 2011; Clune et al., 2009; Gomez et al., 2008). Other studies have focused on optimizing morphological components, such as the caudal fin of a robotic fish (Clark et al., 2012) and flexible joints in terrestrial robots (Moore and McKinley, 2012). As noted earlier, the evolutionary computation study conducted by Hase et al. (Hase et al., 2004) found that 2D animats with simulated neuromuscular morphologies were capable of both bipedal and quadrupedal hopping motions similar to their respective biological counterparts. By applying evolutionary approaches to the study of bipedal hopping in 3D animats, we hope to gain insights into this behavior at a level not previously explored.

## Methods

We began our study with an animat based roughly on the morphology of a kangaroo rat, whose gaits have been analyzed extensively with the aid of high-speed, high-resolution video cameras (Gutmann et al., 2013); see Figure 2. We first evolved gaits for fixed morphologies, then allowed evolution of morphological parameters such as limb dimensions, joint output potential and mass distribution.

**Virtual Animat.** Figure 3 shows the initial animat constructed in the Open Dynamics Engine (Smith, 2012), with body part dimensions corresponding to that of the kangaroo rat. The animat also features a controller that actuates all joints. Kinematic data of the kangaroo rat's hopping gait indicated that the individual joints move in a periodic motion similar to a sine wave. Hence, for this initial study where we focus on steady state hopping gaits, a relatively simple sinusoidal controller was implemented; our ongoing investigations use more complex neural-based controllers. In addition, left/right symmetry was enforced. This decision was made primarily due to the difficulty in evolving a controller for a predefined morphology (unlike nature, where they evolved together). Preliminary experiments found that asymmetric controllers had difficulty achieving stable gaits due to large differences in the length of hind and fore limbs. Moreover, observation of kangaroo rats demonstrates left/right symmetry during hopping.

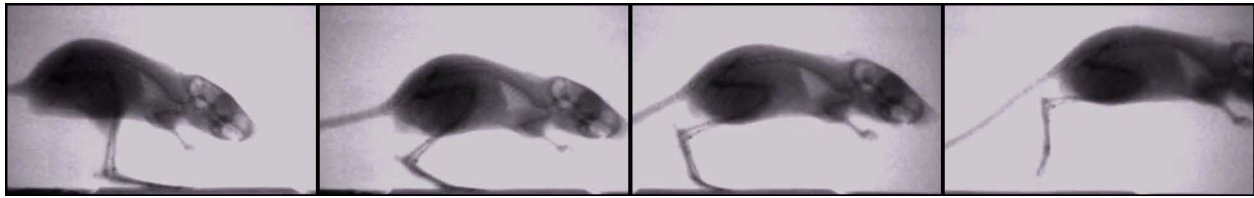


Figure 2: X-ray video progression of a kangaroo rat hopping across a force plate to quantify hopping dynamics.

**Muscle Model.** Animals exhibit fluid movements produced by muscles contracting and relaxing in a coordinated manner. To approximate such dynamics in a rigid-body simulator such as ODE, we modeled muscular connections using hinge joints with appropriate constraints. In particular, we devised a model in which the energy an individual joint can expend during actuation is limited. Doing so prevents situations in which a joint can move with an infinite amount of force, an impossibility in biological organisms. Figure 4 shows the range of motion and relative power of each joint in the morphology. Limiting the maximum force an individual joint can exert produces a system in which multiple joints must work together to move the animat. This muscular model is applied only to the rear legs, as the fore legs do not factor heavily into the locomotion pattern for evolved individuals.

We found that this model produced coordination among the components of the body and natural looking gaits. During locomotion, animal joints do not always move throughout their entire range of motion (for example, strides may be shortened to handle rough terrain, or the center of gravity may be lowered by crouching to improve balance). If the potential were unlimited, joints would always move throughout their full range of motion, irrespective of external forces. By limiting potential, the range of motion of one joint would be indirectly determined by the evolved muscle output parameters of other joints. Moreover, limiting the overall output potential of each joint allowed the limbs to flex and react to the ground when landing, increasing stability and the “naturalness” of the gait.

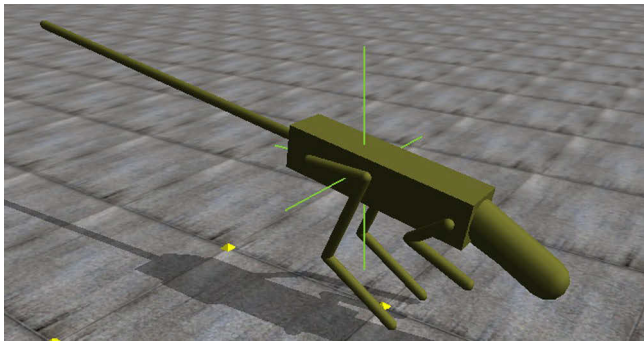


Figure 3: Initial simulated animat used in this study, with morphological dimensions and mass based on kangaroo rat.

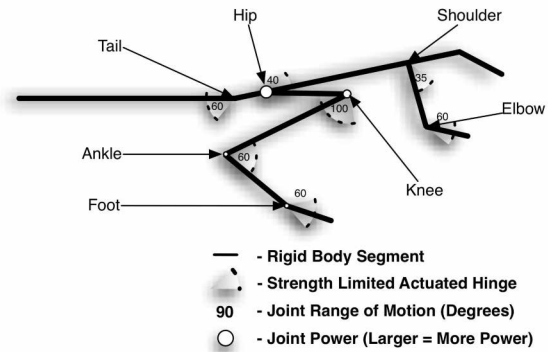


Figure 4: Two-dimensional representation of the animat joints, with range of movement indicated.

**Evolutionary Setup.** For each of five treatments, described in the next section, we executed 25 replicate runs, each with a unique random number seed. In each run, a population of 150 individuals evolved for 4000 generations. Fitness was defined simply to be the distance traveled in 10 seconds of simulated time. No special selective pressure was applied to prefer hopping to other forms of locomotion. Successive generations were populated using 2-way tournament selection with mutation and crossover as defined below. The genome comprised 12, 14, or 16 values, depending on the treatment, as shown in Table 1. For treatments 1 and 2, the genome did not include parameters for an actuated tail.

The mutation rate was relatively high, 20%, but mutations were defined according to a gaussian distribution, so an individual mutation was unlikely to produce a large change in value. We found this approach to be effective given the control strategy used, where a large change in a single key parameter, such as a phase offset, often produced an unstable solution. A more conservative mutation approach allowed for gradual change to gait patterns over generations.

Single-point crossover was applied with a probability of 25% per genome. Crossover exhibited spatial locality, in that parents for an individual solution were chosen within a defined range. Specifically, we applied a geographical approach (Spector and Klein, 2006), where the population is considered as a one-dimensional line with wrap-around. Individuals are produced from parents that are considered to be close to their offspring.

Table 1: Individual Gene Limits

Parameter	Min. Value	Max Value
Actuation Freq.	0 Hz	2.5 Hz
Hip Orientation	0°	337.5°
Knee Orientation	0°	337.5°
Ankle Orientation	0°	337.5°
Toe Orientation	0°	337.5°
Shoulder Orientation	0°	337.5°
Elbow Orientation	0°	337.5°
Center of Mass	body center - $0.25 \times \text{length}$	body center + $0.25 \times \text{length}$
Hip Power	0 (passive)	1.0
Knee Power	0 (passive)	1.0
Ankle Power	0 (passive)	1.0
Toe Power	0 (passive)	1.0
Treatments 3, 4 and 5		
Tail Actuation Freq.	0 Hz	2.5 Hz
Tail Orientation	0°	337.5°
Treatment 5 Only		
Tail Length	$0.07 \times \text{body length}$	$2.2 \times \text{body length}$
Tail Mass	$3.25 \times 10^{-4} \times \text{body mass}$	$0.6 \times \text{body mass}$

## Experiments & Results

The 5 treatments, described below, investigate the role of the tail in bipedal hopping, including interaction with other aspects of the morphology and effect on gaits. To assist the reader in visualizing evolved behaviors, we have placed videos of selected evolved behaviors on a YouTube account:

Treatment 1: <http://y2u.be/V56Xmgf7pxE>

Treatment 2: <http://y2u.be/M1BWxwVUAEM>

Treatment 3: <http://y2u.be/bizIMorOv9g>

Treatment 4: <http://y2u.be/dIyoE0eMm2A>

Treatment 5: <http://y2u.be/OXIbXrwXU3Y>

**Treatment 1: No Tail.** In Treatment 1, individuals lack a tail. Most (18) of the 25 replicate runs failed to produce bipedal hopping, instead evolving *bounding* gaits, where fore and hind limbs alternate contact with the ground. Such gaits were common throughout the study, since they offer relatively stable locomotion, albeit slower than bipedal hopping. Six of the replicate runs were able to manage two or three hops before settling into a forward-leaning gait and then regressing to a bounding gait. However, in one run, the dominant individual, shown in Figure 5 and the Treatment 1 video, exhibited a fairly effective bipedal hopping gait, although it flipped over near the end of the simulation period. Presumably, the bounding gait was a more stable configuration for tailless animats. Examination of early generations found that many individuals attempting to hop tended to flip

over backwards, resulting in low fitness scores. One encouraging trend that emerged in this and subsequent treatments was the effectiveness of our muscle model in simulating flexible joints. During locomotion, joints flexed to react to contact with the ground, resembling the function of biological musculoskeletal systems.

**Treatment 2: Fixed, Rigid Tail.** In the second treatment, individuals had a fixed, rigid tail, and were able to evolve hopping gaits with relatively high fitness values. However, we observed that the majority of successful hoppers used the tail as a “kickstand” to prevent flipping over, as had occurred in Treatment 1. The increased stability enabled individuals to hop farther. The best evolved individual for this treatment can be seen in Figure 5. Most of the replicate runs produced individuals that used their tail in this manner through the entire simulation period, however, a few managed to execute two or three hops between tail taps. Although not ideal, this tail-tapping motion turned out to be an important aspect in the emergence of hopping gaits.

**Treatment 3: Actuated Tail.** The fixed tail in Treatment 2 approximates the initial posture of a kangaroo rat at the start of a hopping motion. In Treatment 3, we expanded the genome to allow the tail to evolve a speed of oscillation value as well as a starting position. We expected to see hopping gaits that did not use the tail as a kickstand as had occurred in Treatment 2. Evolved solutions for this treatment did tend to favor oscillating tails that counteracted the angular momentum of the body. However, the kickstand effect was still present in many individuals, although not as predominant as those evolved previously. In addition to the kickstand function of the tail, evolved individuals demonstrated a coupling between tail and leg oscillation that has the tail moving against the legs to limit the rotation of the body during the hop. An evolved individual for this treatment can be seen in Figure 6, which shows the use of the actuated tail to stabilize the body pitch.

**Treatment 4: Tail Collision Removal.** In a natural environment, hopping species tend not to drag their tails on the ground or even allow the tail to contact the ground at high speeds, in order to avoid injury. In Treatment 4, we explicitly removed the kickstand effect by simply preventing the tail from interacting with the ground. (Effectively, the tail could contact and penetrate the ground with no effect on the animat.) We expected solutions to instead use the tail as a counterbalance to angular momentum, consistent with a prevailing hypothesis in biology (Bartholomew and Caswell, 1951; Alexander and Vernon, 1975; Libby et al., 2012). Instead, the results from all replicate runs tended towards bounding gaits similar to those in Treatment 1. We suspect that the additional mass associated with a tail made it more difficult for the individuals to maintain balance, resulting in the tendency to lean forward.



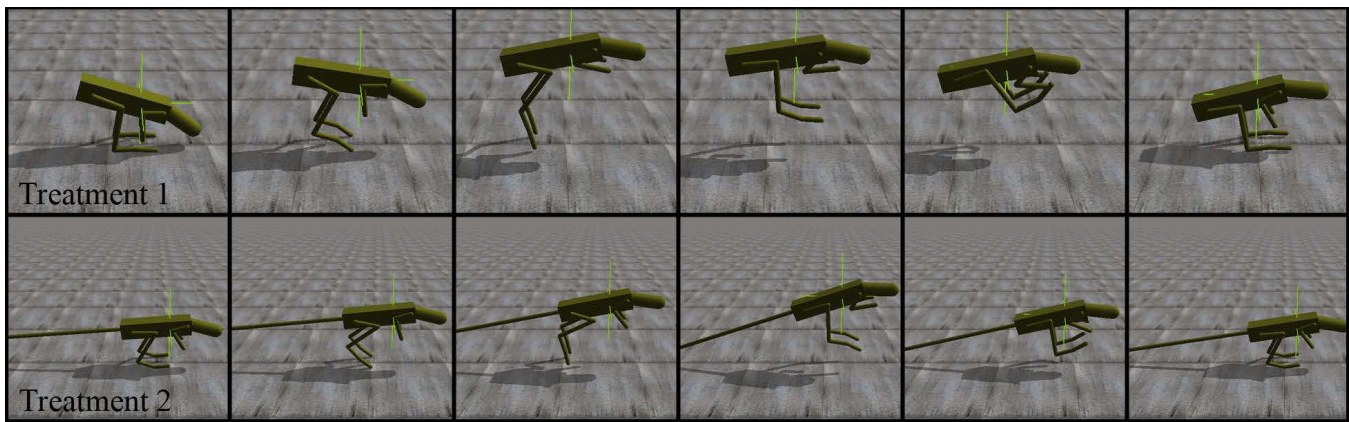


Figure 5: Behavior of evolved tailless and fixed-tail individuals. The fixed tail individual was able to hop more effectively by using its tail as a stabilizer to prevent flipping over backwards.

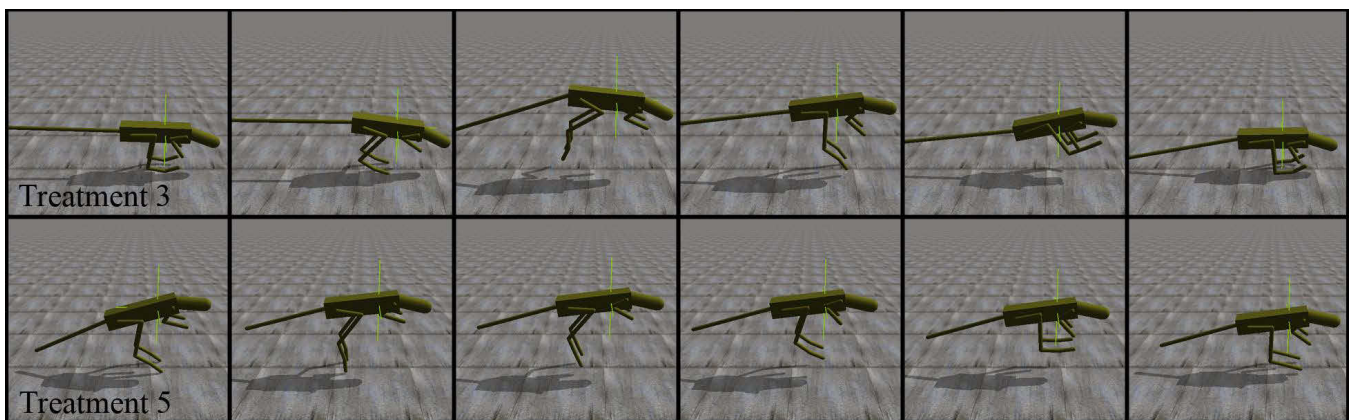


Figure 6: An evolved hopping individual from Treatment 3 with an actuated tail. Note the coordination between tail and legs to maintain body pitch throughout the hopping motion. In the evolved individual from Treatment 5, the tail evolves to be shorter than those of the previous treatments, enabling faster hopping.

**Treatment 5: Evolvable Tail Morphology** In the first four treatments, tails appeared to be essential to maintaining stability. In biology, it is generally agreed that an important function of the tail is to counter the angular momentum of the body, discouraging body pitch changes over the hopping period (Bartholomew and Caswell, 1951; Libby et al., 2012). Since we had based the animat’s morphology on the kangaroo rat, we were curious what solutions would be discovered if tail length and tail mass were allowed to evolve. Indeed, Treatment 5 runs produced bipedal hoppers with tails approximately half as long as those in the earlier treatments; an example is shown in Figure 6.

**Performance Comparison.** Figure 7 plots the best and average fitness for each of the 5 treatments. In Treatment 1, solutions were forced to focus heavily on stable locomotion rather than maximizing the speed of movement, resulting in low fitness. Treatment 4 exhibited even worse performance

in both plots, demonstrating that in these experiments tail tapping is an important part of the behavior, at least as the animat starts moving. Treatment 5 had the best performing individuals across all treatments, although the average performance was similar to that of Treatment 3. This result is likely due to individuals that were unstable and attained low fitness scores. Individuals in Treatment 2 had the second best performance, presumably by using the tail to stabilize the animat during hopping. Treatment 2 also had the best average fitness, indicating that the static nature of the morphology likely made finding stable solutions easier.

**Analysis.** Considering the high performance achieved in Treatment 5, we sought to determine which factors and relationships gave rise to effective bipedal hopping. We discovered that in the top 10% of evolved solutions in this treatment, there existed a relatively tight coupling between tail and leg oscillation frequencies. Figure 8 presents these data

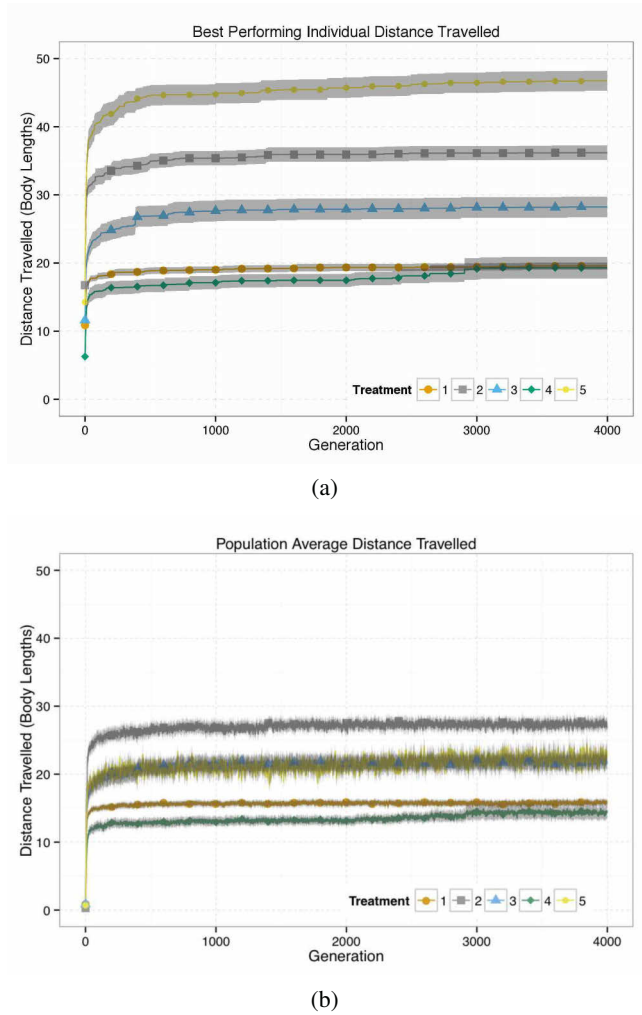


Figure 7: Fitness of 5 treatments over evolutionary time: (a) Best performing individual, averaged across 25 runs for each treatment; (b) Average performance in each evolved population, averaged across 25 runs for each treatment. Shaded bands indicate 95% confidence intervals.

for individuals in the final generation. In the figure, the tail oscillation frequencies are generally near either a harmonic of the leg oscillation frequency, or they act as a passively flexible joint (lower right). Results that fall on or near these harmonic values have tails that move directly opposite to the rotation of the body, apparently helping to maintain a more effective body orientation. In the solutions indicated as passively flexible, the tail oscillation frequencies are so low that they behave as a flexible joint that moves only in reaction to the hopping motion, thus countering rotational movement. The coordination in phase between tail and leg movement is essential for successful individuals and is supported by biological observation. In hopping species, tails tend to move in concert with the rest of the body producing a unified gait pattern. In our observations of evolved animats, individu-

als lacking this coordination tend to produce extraneous or detracting movements that actually hinder performance.

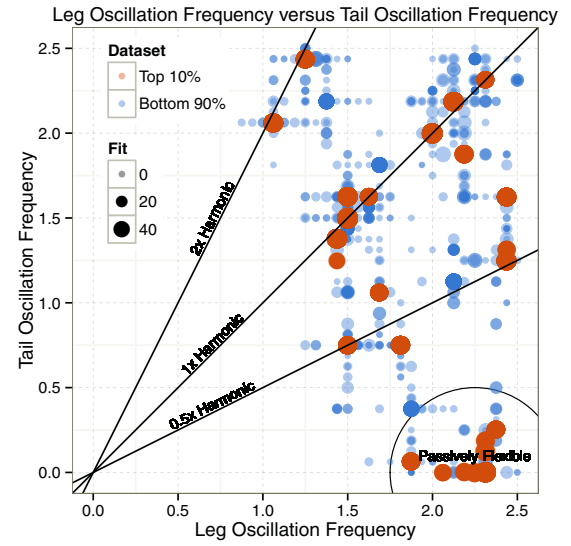


Figure 8: Relationship between the leg oscillation frequency and tail oscillation frequency in Treatment 5. The straight lines indicate harmonics between the two frequencies. Evolved solutions tended to either fall near these lines or in the passively flexible region.

A second area of interest is the evolved mass of the tails and the resulting moments of inertia. As seen in Figure 9, the evolved results tended towards tail masses that were less than 15% of the total body mass. Indeed, tails in some of the best performing individuals accounted for less than 5% of total body mass. These light tails resulted in relatively low moments of inertia, as seen in Figure 10. Lower moments of inertia in these individuals potentially allow the body to generally change pitch throughout the hopping motion rather than maintain a stable body orientation.

This result is intriguing because stable orientation in hopping species benefits from a *high* moment of inertia in tails (Usherwood and Hubel, 2012). Moreover, Figure 10 indicates that there is no direct relationship between the tail moment of inertia and leg oscillation frequency. A possible explanation is related to our evaluation period. While the insight into high moments of inertia for the tails is well understood, the biological observations leading to this conclusion generally focus on *steady-state* hopping. However, in our treatments, fitness evaluation begins at the start of the simulation period which includes the startup phase. Hence, individuals begin from a stationary starting position and must begin to hop before reaching their final steady state. The inclusion of the startup period places an emphasis on stability during the transition from stationary pose to hopping to avoid falling over or becoming unstable. This pressure

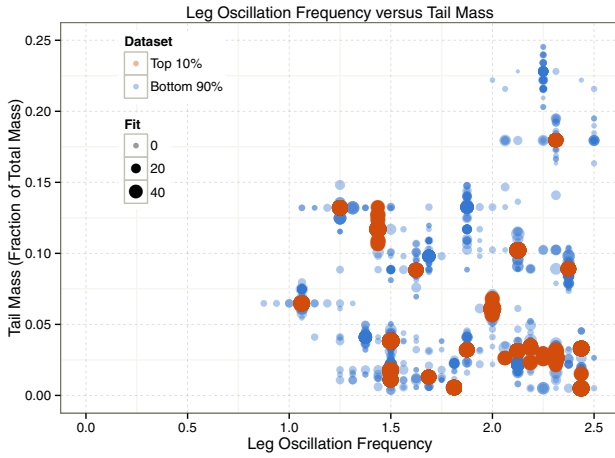


Figure 9: Relationship between the leg oscillation frequency and tail mass as a percentage of total body mass. Lighter tails are favored, although the evolved tail length remains relatively constant even for different masses.

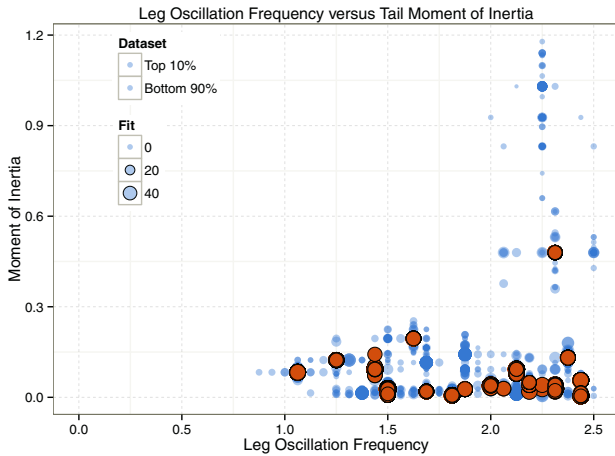


Figure 10: Relationship between the leg oscillation frequency and moment of inertia for an individual. A low moment of inertia generally means the animat is likely to change body pitch during hopping.

likely forces the solutions to evolve parameters that encourage stable startup gaits over those that are most efficient or fastest during the steady-state phase. One possible approach is to delay the evaluation until the animat has had an opportunity to start moving. Adding such a transient phase, which has proven successful in other recent studies (Moore et al., 2013), may encourage tail parameter evolution towards steady state hopping. However, we note that at the time of this writing, a preliminary set of experiments showed that a transient phase actually *reduced* fitness. This issue is a topic of our ongoing research.

## Conclusions

Although relatively uncommon in the animal kingdom, bipedal hopping provides benefits both for energy efficiency and as a survival mechanism. A better understanding of this behavior, and how it evolved, not only informs biology but has implications for the design of robotic systems. We have developed a computationally-efficient kinematic model that approximates the function of natural muscles and is suitable for integration into evolutionary algorithms. In 5 treatments, we explored the role of the tail in hopping gaits. We found that a tail is essential to hopping, as tailless individuals resorted to bounding or shuffling gaits. Evolved gaits exhibit similarities to their biological counterparts in terms of tail movement and joint coordination. However, our results also show that bipedal hopping is not limited to the morphological configurations observed in nature, but can evolve in other morphologies (i.e., those with short, light tails). Indeed, the initial morphology based on the kangaroo rat dimensions proved not to be the most effective morphology. Finally, the inclusion of the startup phase in fitness evaluation led to an alternate use for the tail as a stabilizer, which to our knowledge has not been previously reported.

In future work we plan to conduct more in-depth study of transient versus steady-state hopping behaviors and the pressures influencing them. We also plan to refine our simulated muscular model to more accurately capture the behavior of natural muscle and tendon systems. Finally, as with our studies of aquatic robots, we intend to evolve more complex controllers, using artificial neural networks, for bipedal hoppers.

## Acknowledgments

The authors gratefully acknowledge the BEACON Center for the Study of Evolution in Action, which directly supported this study and fosters an environment for interdisciplinary collaboration. This research also has been supported in part by U.S. National Science Foundation grants CNS-1059373, CNS-0915855, CNS-0751155, CCF-0820220, and DBI-0939454.

## References

- Aguilar, J., Lesov, A., Wiesenfeld, K., and Goldman, D. (2012). Lift-off dynamics in a simple jumping robot. *Physical Review Letters*, 109(174301).
- Alexander, R. M. and Vernon, A. (1975). The mechanics of hopping by kangaroos (macropodidae). *Journal of Zoology*, 177(2):265–303.
- Bartholomew, G. A. and Caswell, H. H. (1951). Locomotion in kangaroo rats and its adaptive significance. *J. Mamm.*, 32:155–169.
- Berkemeier, M. and Fearing, R. (1998). Sliding and hopping gaits for the underactuated acrobat. *IEEE Transactions on Robotics and Automation*, 14(4):629–634.



- Biewener, A. A. and Blickhan, R. (1988). Kangaroo rat locomotion: design for elastic energy storage or acceleration? *J. Exp. Biol.*, 140:243–255.
- Bongard, J. C. (2011). Morphological and environmental scaffolding synergize when evolving robot controllers. In *Proceedings of the 2011 ACM Genetic and Evolutionary Computation Conference*.
- Brooks, R. A. (1992). Artificial life and real robots. In *Proceedings of the First European Conference on Artificial Life*, pages 3–10. MIT Press, Cambridge, MA.
- Cintrón, F. and Mutka, M. (2010). Hopping enhanced sensors for efficient sensor network connectivity and coverage. In *2010 IEEE 7th International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 119–126, San Francisco, CA, USA. IEEE.
- Clark, A. J., Moore, J. M., Wang, J., Tan, X., and McKinley, P. K. (2012). Evolutionary design and experimental validation of a flexible caudal fin for robotic fish. In *Proceedings of the 13th International Conference on the Simulation and Synthesis of Living Systems*, East Lansing, Michigan, USA.
- Cliff, D., Husbands, P., and Harvey, I. (1993). Explorations in Evolutionary Robotics. *Adaptive Behavior*, 2(1):73–110.
- Clune, J., Beckmann, B. E., Ofria, C., and Pennock, R. T. (2009). Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *Proceedings of the IEEE Congress on Evolutionary Computing*, pages 2764–2771, Trondheim, Norway.
- Dawson, T. J. and Taylor, C. R. (1973). Energetic cost of locomotion in kangaroos. *Nature*, 246(5431):313–314.
- Gomez, F., Schmidhuber, J., and Miikkulainen, R. (2008). Accelerated neural evolution through cooperatively co-evolved synapses. *Journal of Machine Learning Research*, 9:937–965.
- Gutmann, A. K., Lee, D. V., and McGowan, C. P. (2013). Collision dynamics of bipedal hopping. In *Annual Meeting of the Society for Integrative and Comparative Biology*, San Francisco, California, USA.
- Gutmann, A. K., McKinley, P. K., and McGowan, C. P. (2012). A detailed musculoskeletal model for studying the evolution of bipedal hopping. Bio/computational Evolution in Action Consortium Congress, East Lansing, Michigan, USA.
- Hase, K., Khang, G., and Eom, G.-M. (2004). A simulation study on the evolution of hopping motions in animals. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3):353–362.
- Ijspeert, A. J. (2001). A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biological Cybernetics*, 84(5):331–348.
- Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4):642–653.
- Jusufi, A., Goldman, D. I., Revzen, S., and Full, R. J. (2008). Active tails enhance arboreal acrobatics in geckos. *Proceedings of the National Academy of Sciences*, 105(11):4215–4219.
- Libby, T., Moore, T. Y., Chang-Siu, E., Li, D., Cohen, D. J., Jusufi, A., and Full, R. J. (2012). Tail-assisted pitch control in lizards, robots and dinosaurs. *Nature*, 481(7380):181–184.
- Moore, J. M., Clark, A. J., and McKinley, P. K. (2013). Evolution of station keeping as a response to flows in an aquatic robot. In *Proceedings of the 2013 ACM Genetic and Evolutionary Computing Conference*, Amsterdam, Netherlands. ACM.
- Moore, J. M. and McKinley, P. K. (2012). Evolving flexible joint morphologies. In *Proceedings of the 2012 ACM Genetic and Evolutionary Computing Conference*, Philadelphia, Pennsylvania, USA. ACM.
- Niiyama, R., Nagakubo, A., and Kuniyoshi, Y. (2007). Mowgli: A bipedal jumping and landing robot with an artificial musculoskeletal system. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pages 2546–2551, Roma, Italy.
- Sims, K. (1994). Evolving virtual creatures. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, pages 15–22.
- Smith, R. (2012). Open Dynamics Engine. Manual and source code available online at: <http://www.ode.org>.
- Spector, L. and Klein, J. (2006). Trivial geography in genetic programming. In Yu, T., Riolo, R., and Worzel, B., editors, *Genetic Programming Theory and Practice III*, volume 9 of *Genetic Programming*, pages 109–123. Springer US.
- Usherwood, J. R. and Hubel, T. Y. (2012). Energetically optimal running requires torques about the centre of mass. *Journal of The Royal Society Interface*, 9(73):2011–2015.
- Zhao, J., Yang, R., Xi, N., Gao, B., Fan, X., Mutka, M. W., and Xiao, L. (2009). Development of a miniature self-stabilization jumping robot. In *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, pages 2217–2222, St. Louis, MO, USA. IEEE Press.



# The role of arousal in embodying the cueXdeficit model in multi-resource human-robot interaction

Kiril Kiryazov<sup>1</sup>, Robert Lowe<sup>1</sup>

<sup>1</sup>Interaction Lab, University of Skovde, Sweden  
kiril.kiryazov@his.se

## Abstract

In this paper is investigated the problem of managing limited resources in human-robot interaction with a computational architecture of emotion. The architecture is based on the appraisal theory of affect and an ethological motivational model of task selection. Key variables and performance criteria for robotic energy autonomous behaviour in interaction with humans are discussed. The role of arousal for modulating effort of movement is explored. It is shown that the architecture can manage task selection and the effort of the movement and offers sustainable basic-cycles in exemplar “two-resource problem” test-bed scenarios for an iCub robot. An extension of the architecture with a third ‘resource’ – safety – is presented and how the architecture is able to solve the new ‘three-resource’ problem is demonstrated.

## Introduction

Research on robots interacting with humans perennially arouses interest. There are many research agendas focused on human-robot interaction (HRI), for example, concerning service robots for helping the elderly or disabled people, robot toys for treatment of autism in children, artificial pets for entertainment, autonomous vehicles and robotized spacesuits. The requisite artificial agents should be able, autonomously, to balance their internal needs such as energy homeostasis, hardware integrity, temperature balance, and at the same time fulfil designer requirements whilst interacting efficiently with the human inter-actor. If a certain degree of autonomy is missing and there is no remote control operation, the agent will require a predesigned action set for every possible environment state, an impossible feat in a complex environment where humans reside.

McFarland suggests that at the root of autonomous control is energy autonomy (McFarland & Spier, 1997; McFarland, 2008). The robot should be able to produce high work quality whilst efficiently utilizing the available energy level of its batteries. Furthermore, the autonomous robot must find the behavioral trade-off between re-charging and working. Managing these two conflicting requirements fundamentally, constitutes the “two-resource problem”, which has become a test bed for several studies in autonomous robotics (Avila-García & Cañamero, 2004; Lowe et al., 2010). The main emphases of the proposed solutions are: firstly, the amount of time the robot should spend at each resource, and secondly, the time at which (work, fuelling) behavioural activities should be switched (sequencing of behaviors) (Wawerla & Vaughan, 2009). Effort during movement is another crucial factor regarding energy autonomy. Responding to urgent situations such as items falling from surfaces, fire hazards,

boiling, or/and overflowing, water requires fast and effortful action. However, in moments of relative calm, the autonomous robot should conserve its energy and, essentially, ‘take it easy’. More effortful movements could raise the safety hazard and this is another reason that the robot should “spare its effort” in situations other than those requiring urgency.

For the purpose of addressing the above considerations for robot autonomy we have developed an affective-cognitive architecture. This combines ‘top-down’ processes, enabling (human-like) expression of emotional state, with ‘bottom up’ processes grounded in the energy balancing mechanisms of the robot. We postulate that this hybrid approach offers the robot greater potential for flexibility when performing tasks that require the cycling of two or more activities. In (Kiryazov, Lowe, Becker-Asano, & Ziemke, 2011) a similar architecture was presented but implemented in a simulated NAO robot. Robot ‘work’ concerned an abstract behavioural sequence of reaching and then stepping onto a certain square on the ground. In the current set of experiments the work is less abstract and represents a more realistic service robot task subcomponent – robot tracking/monitoring of human action. The robot is required to track a ball, which the human inter-actor holds and moves. Specifically, the robot is required to uncover an interactive dynamic in relation to its energetic and movement capabilities guiding viable task (ball tracking, (re)charging) switching behaviour. This is further constrained by a requirement to interact ‘safely’.

The rest of the paper breaks down into the following sections: 2) The two-resource problem and humanoid service robotics – we provide a description of a general two-resource framework for autonomous robots and its adaptation to the proposed service robotics specific requirements; 3) An affective embodied architecture – presenting a cognitive architecture for solving the specified problems; 4) Methodology; 5) Results; 6) Discussion – a summary of present work and insights, and proposed future experiments using the cognitive architecture incorporating emotional expression and recognition mechanisms; 7) Conclusion.

## The two-resource problem and humanoid service robotics

Two-resource problem scenarios offer a common test bed for studying behaviour cycling of autonomous robots under resource constraints. The two-resource problem requires an agent to maintain the level of two basic internal variables in relation to relevant environmental resources. In order for the problem not to be trivial, the robot behaviour that leads to an

increasing satisfaction of one resource-related variable should, most of the time, lead to the inability to satisfy the other resource-related variable(s). In such a framework one of the main problems is the choice of the best action for collecting the right resource at the right moment in time (Wawerla & Vaughan, 2009). The way of gathering a resource is also important as it is stated in (Spier & McFarland, 1986) it is not only important “what to do next” but also “how to do it”. When the two resources are the work and energy, which a mechanical robot should balance, then the choice of an effort of a movement (speed, stiffness) contribute importantly to this two-resource problem.

In the two-resource problem framework of McFarland and Spiers (McFarland & Spier, 1997) the artificial agent (animat) is represented as a state space. The animat is characterized with a minimum set of state variables that are sufficient to describe its psychological state. The animat should maintain homeostasis of this state (not going into a lethal limit), handling the perturbations caused by its actions and the environment. Within this framework, McFarland and Spiers’ experiment with different decision strategies for the animat. They show that very simple cueXdeficit motivation leads to good performance in a lot of different environments. The motivation of a robot/animat to approach a particular resource ( $r$ ) is calculated as:

$$M_r = D_r \times C_r \times K_r \quad (1)$$

where  $D_r$  is the deficit of the resource-related variable in relation to a homeostatic value set by the designer.  $C_r$  is the cue of resource  $r$  and represents its ease of access, e.g. for a mechanical robot a metric for the cue could be the euclidean distance of the resource of the current position of the robot. If a resource is temporarily not visible, instead of having a zero cue value (cf. (Spier & McFarland, 1996)), it is proposed to assign a low fixed value to the cue – the so-called “ambient cue”. This represent “the knowledge” of the agent that there should be some of this resource somewhere in the environment and could motivate the agent to search for it if this resource deficit is very low (even if the agent cannot see it). When the motivation is dependent on the cue, two types of behavioural patterns – opportunism and persistence – emerge, which are important for behavioural stability in two-resource problem frameworks (Avila-García & Cañamero, 2004; Spier & McFarland, 1996).  $K_r$  is the availability of tools in the environment, which are required to handle resource  $r$ . The cue of a resource is based on sensor input and the deficit is measured through proprioception. This model captures animal behaviour at an abstract level where a basic requirement of populations of animals is to choose between food and water resources (Sibly, 1975). For animals the most important resources are food and water or, alternatively, fuel and mating. The work and energy are suggested as robotic analogues for the two basic resources (McFarland & Spier, 1997).

A basic measure for behavioural stability for a robot in such two-resource problem scenarios proposed in the above-mentioned studies is the ability of the system to produce sustainable basic cycles – cycles in the state space of the robot which don’t cross the designer-specified “lethal” limits of the resource-related variables.

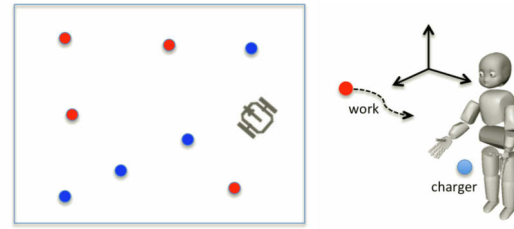


Figure 1. Left: A typical two-resource problem in a 2D world where the robot collects multiple resources of two different types (e.g. blue = water, red = food). Right: A two resource problem for service humanoid robots, where there is one fixed energy resource and a dynamic “work” resource.

For a simple wheeled robot the movement towards a resource and its effort is simply controllable by the speed of the two motors for the left and right wheel. Humanoid robots have many more degrees of freedom and the actions they perform are much more complex. In order to simplify the optimization problem for such robots an algorithmic distinction between the task switching mechanism and the effort of movement is reasonable. In order to translate simple wheeled robot two-resource problems to humanoid robot and service robotics scenarios certain factors should be considered. In common, there is only one “energy resource” and that is a charger, which may be in a fixed position the location of which is knowable regardless of the robot’s current position. However, “work” – which is often related to human behavior, entails complex environmental and social dynamics where interaction with human decision making is concerned. At any time the robot should choose the most appropriate action – for work or charging. The actions can be performed at different speeds or degrees of stiffness of the actuators, the latter relating to precision of performance, safety and energy efficiency.

Robots can have different profiles of energy consumption related to their speed of movement. If movements at low speed are more energy efficient, then the robot has to choose the speed apt to solve the trade-off between energy and work efficiency. If the robot consumes less energy for higher movement speed, then the robot basically should move at the highest possible speed for optimal behaviour. However, as previously suggested, higher speed could raise difficulties in HRI, e.g. decreased safety.

Safety is an important aspect related to effort of movement in HRI. Higher speed or more power in the actuators will also mean a greater safety hazard in a close human-robot interaction. Safety could be defined as a third resource, in the same framework, which the robot should maintain within appropriate limits in order to be efficient. One way the robot could gather “safety” is to completely stop when the safety hazard assessment is too high (for example, a human is nearby and the robot is moving fast). Another more relaxed way is to reduce speed and stiffness of the motors.

Generally, emotion plays an important role for efficient communication (Thill & Lowe, 2012), (Bar-On & Parker, 2000). Expressing emotion states could provide fast feedback to another humanoid with respect to the current mental state of the partner in a cooperative task and potentially enhance the human-robot interaction. For safety purposes, it is very important that the robot recognizes human intention and action in order to act appropriately. On the other hand, the robot should meaningfully express its intention.

## An affective embodied architecture

The basic schema of the proposed architecture is shown in Figure 2. Part of the architecture is based on WASABI (Becker-Asano, 2008), which is a psychologically plausible appraisal theoretic model that has been successfully implemented and tested in the context of the virtual human MAX, who served as a museum guide and a card game player (Becker-Asano, 2008). In our architecture the unconscious appraisal process of WASABI and emotion dynamics in PAD space are used mainly. The task switching module substitutes for the BDI decision-making module of WASABI with the *cueXdeficit* action selection strategy.

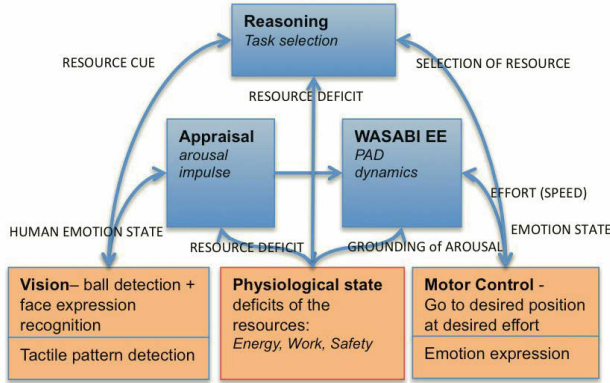


Figure 2. Basic modules of the architecture: modules depicted in orange boxes are robot specific. The other three (in blue) are implementation independent.

A brief description of the components of the architecture, which are relevant to the provided experimental results, follows below. Some of the modules, which are important for emotional communication with human (face expression, face expression recognition, tactile pattern detection), are not relevant for the presented results. However, we envisage their deployment in for future test scenarios. Their role for the efficient management of the resources and possible future experiments are proposed in the discussion section.

### Task switching

The task-switching module uses *cueXdeficit* action selection mechanisms (based on the model of McFarland and Spiers (Mcfarland & Spier, 1997)). The robot chooses an action of accessing a resource depending on which of the resources the robot has higher motivation for. This can be described by the following rule:

Gather resource  $k$ : if  $M_k = \max_i M_i$ , where  $M_i$  is the motivation for choosing a particular resource  $i$ :

$$M_i = C_i \times D_i, \quad (2)$$

$C_i$  – is the cue of the  $i$ -th resource and is provided by the vision module.

$D_i$  – is the deficit of the  $i$ -th resource. It is provided by the physiology module. In addition to the natural physiology measure of the energy level it also includes the other resources – work and safety – which are dependent on the

designer's goal. In this way the robot interprets the social environment in the same way as the physical one as suggested in (Dautenhahn, Ogden, & Quick, 2002)

### Appraisal

The appraisal process is controlled by generating and sending to the WASABI engine emotion impulses in the form:

$$Impulse = K \sum_{i=all\ resources} D_i^2 \quad (3)$$

where  $K$  is a coefficient tuned experimentally according to the dynamics of the Emotion Engine in such a way that the level of arousal during a run of the system could take all of the values from its minimum to maximum value (not remaining at the same level all of the time).

The emotion impulse equations represent general principles of unconscious appraisal in WASABI – that the impulses sent to the emotion engine should represent bipolar evaluations of the environmental state. In the current variant we only use the arousal dimension and negative impulses. From another side, the proposed impulse equation represents the cost function provided in (Mcfarland & Spier, 1997), which is a general criterion for deviation of the animat's equilibrium point in its state space.

### WASABI emotion engine

In WASABI the emotions are represented in the three-dimensional space of pleasure, arousal, and dominance by points with activation and saturation thresholds. The space itself is commonly referred to as PAD space.

The emotional state is a point in this 3D space and has internal dynamics, which is modulated by emotion impulses and other modules in WASABI. In the work presented here, the architecture uses only the arousal dimension. The WASABI engine modulates the execution of the task in the control module. The speed of each movement is made proportional to the current arousal level. A Similar relation between speed and arousal is shown in (Paterson, 2002) where the correlation between the perception of a character's arousal and its speed of movement is studied. WASABI engine also engages face expressions, which are not used in the results but discussed in the last section.

### Vision

The vision module evaluates the cues of the resources based on the perception capabilities of the specific robotic implementation. For the current version cues of energy resource (charger) and work resource (a moving ball) are:

$$C_{work} = \frac{MaxDistanceToWork - distanceToWork}{MaxDistanceToWork} \quad (4)$$

$$C_{energy} = \frac{MaxDistanceToCharger - distanceToCharger}{MaxDistanceToCharger} \quad (5)$$

In this way both cues' values are normalized in the interval [0,1]. As the safety is a special resource without a specific environmental cue representing it, its cue is set to the highest value of 1 all the time. This is also validated from the fact that



gathering this resource – stopping completely – could be achieved at any time immediately.

### Physiological state

The module calculates the resources' deficits based on introspection – monitoring internal physical properties such as electrical currents or more “abstract” deficits such as work and safety, which depend on the designer's requirements.

### Control

This module executes action for gathering the resource whose motivation is the highest, as provided by the task selection module. The action parameters, e.g. for effort of movement, are modulated by the emotion state in the WASABI engine.

The action for the work and charger resources consists of moving the robot's hand towards their position. Safety is “gathered” by stopping at a place for a certain period. The speed of the movement is proportional to the arousal level scaled in such a way that the resultant speed remains within appropriate limits (not too high: undesirable for the real robot because of possible damage; not too low: prohibitively slow for HRI).

## Methodology

For the specific tests of the proposed architecture we are using a scenario where a humanoid robot should follow a ball that is held, in front of it, at random positions by a human. When the robot moves it loses energy and in order to regain it, it should move its hand to a certain “resting” position. This scenario can be viewed as a minimalist analogy to a service robot having to refuel but also carry out a task that involves unpredictable dynamics. This work may also be viewed as a fundamental requirement of robot interaction with a human, i.e. that has some ability to dynamically track the activity of the human. The robot should cycle efficiently between working and refuelling activities in order to remain viable, i.e. not run out of energy, work and safety. The complexity of the task in this case is mainly caused by the “unpredictable” human behaviour – providing complex patterns of the ball (work resource) movement.

The lethal limits for the energy requirements of the robot are obvious – it should never go beyond a (lower) limit of a completely exhausted battery at which the robot stops moving and requires human intervention. In another, not so minimalistic, variant a higher “hibernation” threshold could be implemented which allows the robot to finish what it is currently doing and to find a safe location to ‘sleep’. This hibernation threshold can be manipulated from another “hibernate” behaviour. In the current experiments with the simulated iCub we use energy consumption proportional to the speed of movement. This is the simplest approximation to the real data, which we have collected with the real iCub. With such energy consumption dynamics, higher speed of movement is both more energy- and work-efficient.

The work limits are not obvious as the robot work requirements depend on the goals of the designer (Spier & McFarland, 1986). In the investigation presented in this paper, we use a convenient and natural measurement for the deficit of the work resource – a linear work production decrease from

the time point at which the work is “sensed” (the robot is aware of work to be done but cannot presently do it) and increase when it is actually working. In this way the work resource obtains constrained cyclical continuous dynamics similar to a physical resource. Here this is done more naturally than in (McFarland & Spier, 1997) where the robot should “pay” for energy when charging with its work points.

In order to define the safety resource dynamics we first define:

$$\text{Safety Hazard} = \text{Speed} \times \text{Human proximity} \quad (6)$$

The proportionality of safety to human distance is used in almost every measurement for safety hazards (Calinon, Sardellitti, & Caldwell, 2010; Kulić, 2006). The speed of the robot is a main factor for determining the force in an eventual collision so it is also of crucial importance for defining safety hazard (Kulić, 2006).

Any time the robot moves, it starts to decrease its safety variable level with the current level of safety hazard. When the robot stops at the charger or because it decided to “gather safety” the safety resource increases in the same constant linear fashion as the work resource. The latter two aspects of the safety dynamics were added mainly in order to obtain the same continuous and cyclical dynamics as the other two resources. More reasonable is that safety just equals the safety hazard. This is planned in the further development of the architecture where safety gathering could also relate to continuously decreasing of speed rather than a complete stop as in the current version. Another possible option of safe gathering behaviour could be the manipulation of joint stiffness, for example, decreasing the stiffness when the safety resource motivation becomes dominant. This will also make the change of safety hazard continuous with naturally constrained upper and lower levels.

In the so proposed three-resource problem the choice of right speed of movement is of crucial importance. High speed will decrease the safety. A slowly moving robot, however, will require more time to reach the ball and will reduce the amount of gathered work. A flexible mechanism is required to choose the proper speed at the right time. We suggest the arousal mechanism of our architecture is a suitable modulator of the movement speed for solving the three resource problem and produce sustainable basic cycles.

We are primarily interested in the three-resource problem that consists of managing variables for fuel, work and safety needs. However, the proposed architecture, as a basic requirement, should be able to handle the two-resource problem – produce stable basic behavioral activity cycles. Therefore, we first evaluate whether the cueXdeficit strategy is still applicable in the new setup and explore the role of arousal for modulating the robot's behaviour.

In a second experiment we use safety as a third resource and evaluate whether the architecture can maintain sustainable basic cycles in this three-resource problem. Although, as pointed out before, the safety hazard is defined as a reasonable measure the lethal limits of this resource are arbitrarily chosen. To demonstrate that such eventual behavioural stability is not trivial (the safety resource could be depleted in some of the environments) we will compare the performance of two variants of the architecture:

1. With three resources (work, energy, safety) and
2. With only two – resources (work, energy).

In the latter case we still use the same measure for the safety deficit but the architecture is “blind” for this “safety” resource. Although it can never take the decision to “gather safety” it could still “survive” in this case as the robot stops at the charger and also could stop for a shorter period when it is working (the ball is moving slowly). We predict that even if the robot dies in some of the environments in the “two-resource” variant, it can still be viable in the “three-resource” variant – so the average life-time of the “three-resource” robot-architecture system will be longer.

If there are only two resources, for the current robot energy consumption dynamics, the robot should go as fast as possible so the arousal mechanism is not so important. When the safety as third resource is announced that makes the problem of right speed non trivial because higher speed will increase the safety deficit. It is worth checking, however, whether the robot moving at the highest possible speed can still provide a solution for the three resource problem.

The experiments are performed using 21 runs for each of the two variants of the two architectures in 3 environments (therefore, 7 runs for each environment) for 35000 program cycles (approximately 15 minutes for each run). The detailed description of the used environments is summarized in the annex (table 2). In all environments the ball is moving with a sinusoidal speed on a line in front of the robot. In regular time intervals it disappears. The varied parameters are *the speed of movement of the ball* and *the time it is hidden* which are the key factors for environmental complexity. Higher speed of ball movement will require the robot to move more often and at a higher speed so that it will lose more energy and safety. If the ball is visible for a longer period, that will eventually decrease the work deficit if the robot is not succeeding to track it. The continuous pattern of movement is mainly chosen for implementation reasons – the vision module recognizes the ball much more easily when the ball’s position change is continuous with time (without sudden jumps), which actually should be the case if a human moves it. Although this simulated deterministic environment could look different to the unpredictable human behaviour it still has enough complexity, which for a simple reactive robot is not trivial. Extra complexity comes from the noise of vision and control modules.

For measuring the agent viability, we use two of the proposed viability indicators in (Avila-García & Cañamero, 2004) – life span and overall comfort:

$$LifeSpan = t_{life}/t_{run} \quad (6)$$

where  $t_{life}$  is the number of steps when the robot is viable,  $t_{run}$  is the maximum number of steps of one run

$$OverallComfort = \sum_{i=1}^{t_{life}} (1 - \bar{d}_i) / t_{life} \quad (7)$$

where  $\bar{d}_i$  is the mean of the deficit of all resources

The particular robotic implementation used for the experiments here uses the iCub robot (Sandini, Metta, and Vernon 2007). Most of the data presented here is collected in the iCub simulator. Some demonstrative results with the real robot are shown in the discussion section. For emotion recognition with the simulated robot the video stream of the web-camera instead of the cameras in the eyes of the

simulated iCub is sent to the face emotion expression recognition module.

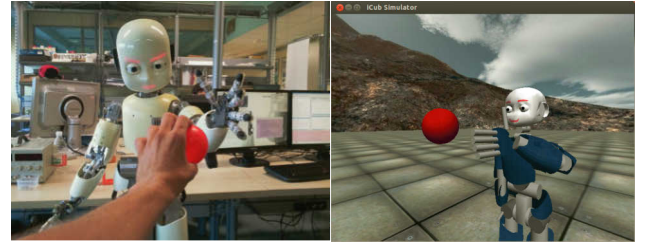


Figure 3 An iCub robot following a red ball – real (left) and simulated (right).

The distance to the work and energy resource is calculated by using the Euclidian distance from the current hand position to the ball and the charger position. There is not any particular physical object representing “the charger” but we assume that the agent knows its state through proprioception.

The following modules from the iCub repository are used for the particular implementation of the vision, physiology and the control modules of the architecture:

- pf3dtracker – detects a single coloured ball and returns its coordinates in a 3D robot-centric reference frame
- ICartesianControl – moves the hand to a desired position for a desired time
- controlBoardDumper – logs the current consumption of the individual motors. Each value is time stamped, so the user can process them offline to compute the instantaneous power consumption
- emotion interface – controls patterns of LED on the face of the robot producing human-like facial expressions.

The last two modules are available only for the real robot and are not relevant for the following results with the simulated robot.

## Results

### Experiment 1: Arousal & the two-resource problem

The robot-architecture system ‘survived’ all the trials – it didn’t cross the lethal limits during the runs in this particular set of environments. That shows that cueXdeficit strategy is still viable strategy in our two-resource problem framework. In order to explore the role of the arousal mechanism we compare one of the runs in a more challenging environment (where the ball is shown for a longer period of time – environment 3 – see annex) and one in the least challenging (environment 1 – see annex).

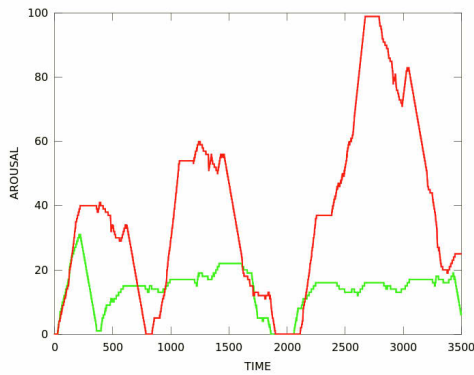


Figure 4. Arousal level of the robot in an “easy” environment – (green) and a “hard” (red). The time is program cycles.

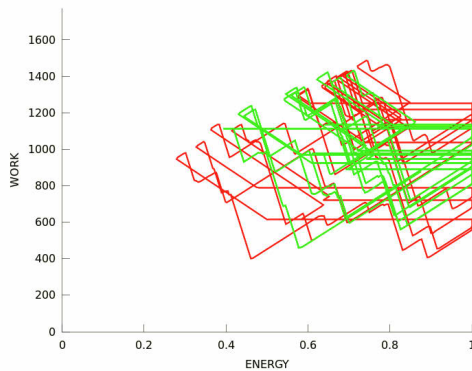


Figure 5. Basic cycles of a robot in an “easy” environment (green) and a “hard” one (red). The axes are also the lethal limits lines for the energy and work resources.

We can see that in the ‘easy’ environment arousal remains relatively low and the corresponding basic cycles are further from the lethal limits than in the more challenging environment where greater arousal is needed to produce behavioural stability.

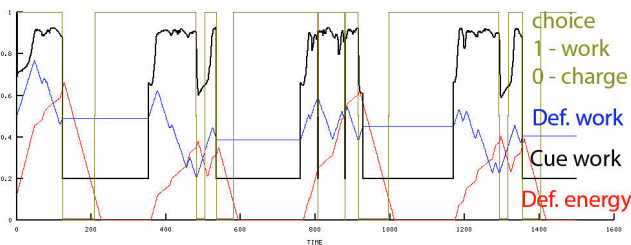


Figure 6. Dynamics of some of the essential variables in a short time window of a run of the architecture in environment 3.

It is interesting to question why the robot doesn’t go into ‘over-opportunism’ (consuming a resource until it dies from the deficit of another resource) in this scenario where the consuming of the resources doesn’t deplete as in ethological experiments with wheeled robots. If a resource is not disappearing, that makes its cue value in the moment of gathering the highest possible value and an agent using a cueXdeficit strategy could become almost blind to the rising

of the deficit of the other resource. But as one can see in Figure 6 the robot still chooses to charge although the ball is present and very close to the hand if the energy deficit becomes considerably higher than the work deficit. One of the factors facilitating this is that the highest received cue value of the work is not maximum (at 1) but a little bit lower which is a side effect of the particular “embodiment” of the architecture (control module accuracy parameters and the dynamics of the tracking motion process). The periods with fixed cue of 0.2 (the ambient cue level) is when the ball disappears. In our scenario the ambient cue could be of benefit in the periods when the ball is missing for a certain short period but there is very high work deficit. In this case, the robot could still stay nearby “waiting” instead of starting to go to the charger. Although the robot is not over-opportunistic it still shows opportunism and persistence - it doesn’t switch to collect the other resource immediately as the deficit drops lower.

## Experiment 2: Safety & the three-resource problem

Firstly, we tested the three-resource variant of the architecture without arousal mechanisms but moving at highest possible speed. The architecture still survived the easy environment (1) but died in all the trials in the harder environments (2 and 3). Secondly, we performed tests with the proposed arousal mechanisms for modulating the speed. The basic cycles in the 3D space of the three resources’ deficit for one of the “viable” runs of the 3-resource architecture are shown in Figure 7. One can observe the cycle’s sustainability – the robot is not crossing any of the three lethal limits planes.

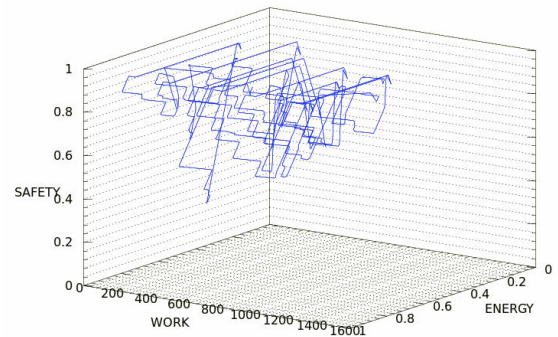


Figure 7. State space trajectory of a run of the “3 resource” system. The three planes 0 values provide lethal limits.

A comparison of the viability indicators of the “two resource” and “three resource” architecture is shown in Figure 8.

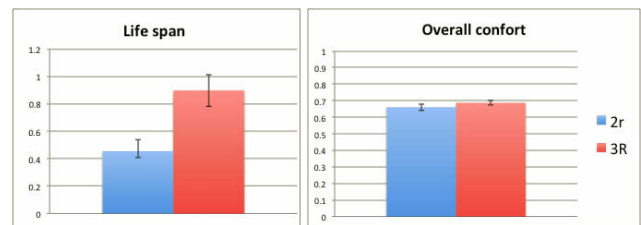


Figure 8. Viability indicators for the architecture with or without safety as the third resource. The graphs show the mean value of all (21) runs and standard error of the mean.



We performed one-tailed t-test in order to compare the life span in the two conditions (as we had hypothesized better performance in the three resource variant) and a two-tailed t-test for the overall comfort. There is no significant difference in the overall comfort -  $p = 0.26765$ . The life span however is significantly different at the 0.01 level of significance ( $p = 0.000235$ ). This result shows that the proposed 3-resource problem is not trivial (the robot dies before the end of the trial in the 2 resource case in 6 of the runs crossing its safety lethal limit) and succeeds in adapting (prolonging its life-time) when the architecture is extended for handling the three resources.

In Figure 9 the average deficit of safety is plotted. A one-tailed t-test for the mean safety shows the three-resource runs' safety significantly increases at the 0.01 level of significance ( $p = 0.000539$ ). This combined with the fact that overall comfort remains relatively the same shows that generally the new architecture compensates the pressure of having a new resource to handle with lowering the efficiency of gathering the other two resources – work and energy.

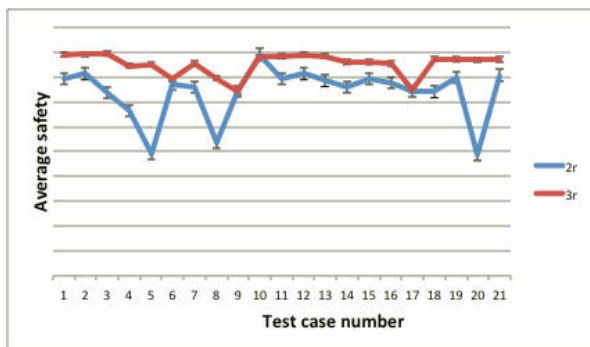


Figure 9. Average safety deficits with standard error for all the test cases

## Discussion

One possible criticism about the presented results with safety is that having such a flexible decision strategy regarding safety is not an appropriate way to deal with such an important issue as safe HRI. Usually industrial robots are programmed to completely stop whenever a human is detected in their workspace. That is a reasonable for robots with rigid bodies, big mass and high speed of movement because danger of injury in any collision between robots and humans is too big and should be avoided at any price. This is not the case however when the robots are “soft”. For example iCub has ImpedanceControl mode, which can be run parallel with CartesianControl. In this mode robot react to any external force and this could dampen a collision with human. The current version of iCub is still made of rigid material but future service robots could be made of soft materials so any possible collision with humans will not be damaging because of the soft robot body. When the robots have such “soft” properties the collision with humans is not something which should be avoided at any cost but could still be an unpleasant experience. Therefore, it could be flexibly balanced with the other requirements such as work and energy efficiency.

One of the main differences of our arousal mechanism with the hormone mechanism used in the competitive two-resource problem in (Avila-García, 2004) is the “inertial” dynamics of

arousal in connection with environmental changes. WASABI engine has its “internal inertia” – the arousing events need little time before the arousal becomes high and more time to cool down. The advantage of this inertia has not been demonstrated here but this could be done in experiments when the “dangerous” (arousal causing) event are not uniformly distributed but are grouped in time. Another advantage of such a property is that it is closer to the arousal dynamics in the biological systems.

There are several studies showing that other people’s emotional states tend to induce similar ones in human interactors (Hatfield, Cacioppo, & Rapson, 1994). A more aroused human when the robot is moving faster and calmer when there is no urgency would tend to increase safety within the interaction. Further experiments, however, with human participants and iCub are needed to validate that assumption. In order to express robots’ emotional states three different sets of LED face patterns corresponding to the current level of arousal are used. In (Paterson, 2002) it is shown that people assess the level of arousal based on the corresponding speed of movement. The emotion expression of the real iCub serves as extra emotional feedback.

From another angle, recognizing the emotional state of the human in a cooperative task could also be of crucial importance. Human arousal could be a fast heuristic of anticipated danger. A robot could detect the arousal and adopt its behaviour appropriately before the human becomes aware of the anticipating event (Rani et al., 2004). Multimodal emotion expression recognition based on the face expression recognition module and touch pattern classification is being developed. The system can distinguish two-different types of facial expressions and three touch patterns in real time. Future experiments are planned which will prove the role of human emotion state recognition benefits for resource managing in HRI.

## Conclusion

In this paper a particular two- and three-resource problem setup for humanoid robots in a scenario suitable for service robotics tasks is reported. In addition to that mainly exploited in other similar studies – task switching – the concept of ‘effort’ is emphasized here, which is essential for two-resource problems and is studied separately from the action selection. A bio-inspired affective architecture, for solving the multiple resource problems, is presented. We show that the proposed architecture preforms stable basic cycles – a measure for behaviour stability for two-resource (work, energy) problem. The role of ‘safety’ as a necessary third importance resource in HRI is presented. The safety and work efficiency although being high-level entities dependent on designer requirements are successfully incorporated in the architecture in the same way as a physically constrained resource such as the battery energy level. We show that with a reasonable safety measure the architecture can handle the three-resource problem. The arousal, used for maintaining the right effort of movement, is an important modulator of the cueXdeficit action selection mechanisms in the two and three resource problems.



## Acknowledgements

This research has been supported by the EU project ROBOT-DOC under G.A. 235065 from the 7th Framework Programme Marie Curie Action ITN (<http://www.robotdoc.org>).

## References

- Avila-García, O., & Cañamero, L. (2004). Using hormonal feedback to modulate action selection in a competitive scenario. *From Animals to Animats: Proceedings of the 8th International Conference of Adaptive Behavior*, 243–252.
- Bar-On, R., & Parker, J. (Eds.). (2000). *The Handbook of Emotional Intelligence: Theory, Development, Assessment, and Application at Home, School and in the Workplace* (p. 544). Jossey-Bass.
- Becker-Asano, C. (2008). *WASABI: Affect Simulation for Agents with Believable Interactivity - Volume 319 Dissertations in Artificial Intelligence (Dissertationen Zur ... in Artificial Intelligence)*. IOS Press.
- Calinon, S., Sardellitti, I., & Caldwell, D. G. (2010). Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 249–254). IEEE.
- Dautenhahn, K., Ogden, B., & Quick, T. (2002). From embodied to socially embedded agents – Implications for interaction-aware robots. *Cognitive Systems Research*, 3(3), 397–428.
- Hatfield, E., Cacioppo, J. T., & Rapson, R. L. (1994). *Emotional Contagion (Studies in Emotion and Social Interaction)*. Cambridge University Press.
- Kiryazov, K., Lowe, R., Becker-Asano, C., & Ziemke, T. (2011). Modelling Embodied Appraisal in Humanoids: Grounding PAD space for Augmented Autonomy. *Proceedings of the Workshop on Standards in Emotion Modeling*.
- Kulić, D. (2006). *Safety for human-robot interaction*. University of British Columbia.
- Lowe, R., Montebelli, A., Ieropoulos, I., Greenman, J., Melhuish, C., & Ziemke, T. (2010). Grounding Motivation in Energy Autonomy: A Study of Artificial Metabolism Constrained Robot Dynamics. *Artificial Life XII: Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*.
- McFarland, D. (2008). *Guilty Robots, Happy Dogs: The Question of Alien Minds* (p. 256). Oxford University Press, USA.
- McFarland, D., & Spier, E. (1997). Basic Cycles, Utility and Opportunism in Self-Sufficient Robots. *Robotics and Autonomous Systems*, 20, 179–190.
- Paterson, H. (2002). *The perception and cognition of emotion from motion*. University of Glasgow.
- Rani, P., Sarkar, N., Smith, C. A., & Kirby, L. D. (2004). Anxiety detecting robotic system towards implicit human-robot collaboration. *Robotica*, 22(1), 85–95.
- Sibly, R. (1975). How incentive and deficit determine feeding tendency. *Animal Behaviour*, 437–466.
- Spier, E., & McFarland, D. (1986). Possibly Optimal Decision-Making Under Self-sufficiency and Autonomy. *Journal of Theoretical Biology*, 189, 189–317.
- Spier, E., & McFarland, D. (1996). A Finer-Grained Motivational Model of Behaviour Sequencing. *From animals to animats 4: proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, 255–263.
- Thill, S., & Lowe, R. (2012). On the Functional Contributions of Emotion Mechanisms to (Artificial) Cognition and Intelligence. In J. Bach, B. Goertzel, & M. Iklé (Eds.), *Artificial General Intelligence SE - 33* (Vol. 7716, pp. 322–331). Springer Berlin Heidelberg.
- Wawerla, J., & Vaughan, R. T. (2009). Robot task switching under diminishing returns, 5033–5038.

## Annex 1

The main parameters of the architecture for the presented results are summarize in the table below:

Table 1 architecture parameters

Parameter	Value
Maximum of work	1770
Maximum of energy	1
Maximum of safety	470
Distance to the ball when “work is consumed”	0.11 m
Time of standing still when “safety is consumed”	3.5 sec
Ambient work cue value	0.2

Ball position is controlled by equations of time shown below. Coordinates of points are in robot-centric coordinate system with a centre on the floor below the legs of the robot. The ball is hidden in regular time patterns in order for the environment to not be too hard to live in.

Table 2 Environment properties (ball movement patterns)

Environment index	Ball movement pattern
1	Ball position = $A + (\cos(t/4)/2 + \frac{1}{2}) * B$ t- current time in seconds A (0,1,0.3), B (0,0.87,0.4) Ball is hidden every 50 sec for 50 sec
2	Ball position = $A + (\cos(t/3.5)/2 + \frac{1}{2}) * B$ t- current time in seconds A (0,1,0.3), B (0,0.87,0.4) Ball is hidden every 50 sec for 50 sec
3	Ball position = $A + (\cos(t/4)/2 + \frac{1}{2}) * B$ t- current time in seconds A (0,1,0.3), B (0,0.87,0.4) Ball is hidden every 60 sec for 40 sec

# Simulating Sleeping Sickness: a two host agent-based model

Simon Alderton, Dr. Jason Noble & Prof. Peter Atkinson

Institute for Complex Systems Simulation  
University of Southampton  
simon.alderton@soton.ac.uk

## Abstract

Agent-based modelling is useful for policy evaluation in fields such as epidemiology. The current paper presents a model of Human African Trypanosomiasis (HAT), or sleeping sickness: a disease which is becoming increasingly prominent due to recent epidemics. Associated medication is often scarce, whilst diagnosis through blood screening is not always effective. Current modelling methodology uses simple reaction-diffusion models to predict future epidemics, but this makes policy at the village level difficult to evaluate. Agent-based, object-oriented simulation provides a simple means of adding complexity to models of sleeping sickness, allowing the easy incorporation of spatial and vector data. We present an exploratory two-host agent-based simulation for humans and cattle, applying known values for sleeping sickness infection rate, before evaluating the model's policy implications and suggesting steps for future improvement.

## Introduction

Agent-based modelling (ABM) in artificial life has long been used to examine fundamental questions in areas such as the evolution of cooperation or communication. However, ABMs have also been used in a pragmatic way in disciplines such as anthropology (e.g. Gumerman et al., 2003; Lansing and Kremer, 1993), conservation biology (e.g. Watkins et al., 2011), and epidemiology (e.g. Muller et al., 2004; Auchincloss and Diez Roux, 2008). In this latter tradition, we will focus here on an agent-based model of Human African Trypanosomiasis (HAT), also known as sleeping sickness. Our goal is to show how an ABM approach can improve on conventional modelling methods in assessing the likely success of different policies for managing the disease.

HAT is a neglected tropical disease (NTD) (Simarro et al., 2010) and one of the most common conditions affecting the poorest 500 million people living in sub-Saharan Africa (Hotez and Kamath, 2009). Sleeping sickness is a vector-borne, parasitic disease which is transmitted to humans by the bites of the tsetse fly from the genus *Glossina*. The disease is caused by protozoa of the species *Trypanosoma brucei* - namely the sub-species *T. b. gambiense* and *T. b. rhodesiense* (Fèvre et al., 2008) and, as of 2005, was responsible

for an estimated 100,000 deaths every year (Picozzi et al., 2005).

In recent years, the potential for a spatial cross-over of the two forms of the disease has been heightened due to the continued northward spread of Rhodesian HAT in Uganda (Batchelor et al., 2009). This would be particularly significant as Gambian and Rhodesian HAT require different diagnosis and treatment, and any overlap would compromise previous disease characterisation based on knowledge of the geographical distributions of the diseases.

The resettlement of communities in response to epidemics in the 1900s and 1940s had aided the control of the disease. However, a large volume of uncontrolled movements through tsetse infested bush following a further large outbreak in 1971, combined with a lack of resources and trained personnel, meant that mitigation efforts were hindered in a time of political and economic unrest (e.g. Matovu, 1982; Okiria, 1985). As a result of the poor control measures, the *T. b. rhodesiense* form of the disease spread northwards towards the Tororo district in 1984 (Mbulamberi, 1989), before reaching the Soroti district on the north west shores of Lake Kyoga in 1998 (Fèvre et al., 2004), and the Kaberamaido district in 2004 (Fèvre et al., 2005).

At present, with cases of *T. b. rhodesiense* being diagnosed as far north as the Ugandan district of Lira, there is a distance of only 150 km separating the active foci of Rhodesian sleeping sickness to the north of Lake Kyogo, and the Gambian form of the disease towards the north west of Uganda and south Sudan (Figure 1).

## The vector

Mitigation techniques for sleeping sickness in sub-Saharan Africa often focus on the reduction or the removal of the tsetse fly, which carries the disease. For example, while the burning of tsetse infested bush has fallen out of favour due to the associated environmental damage, other effective techniques have incorporated community led 'vector trapping' to reduce the concentration of the flies (Joja and Okoli, 2001). The associated research concluded that the integration of public participation aided community learning and

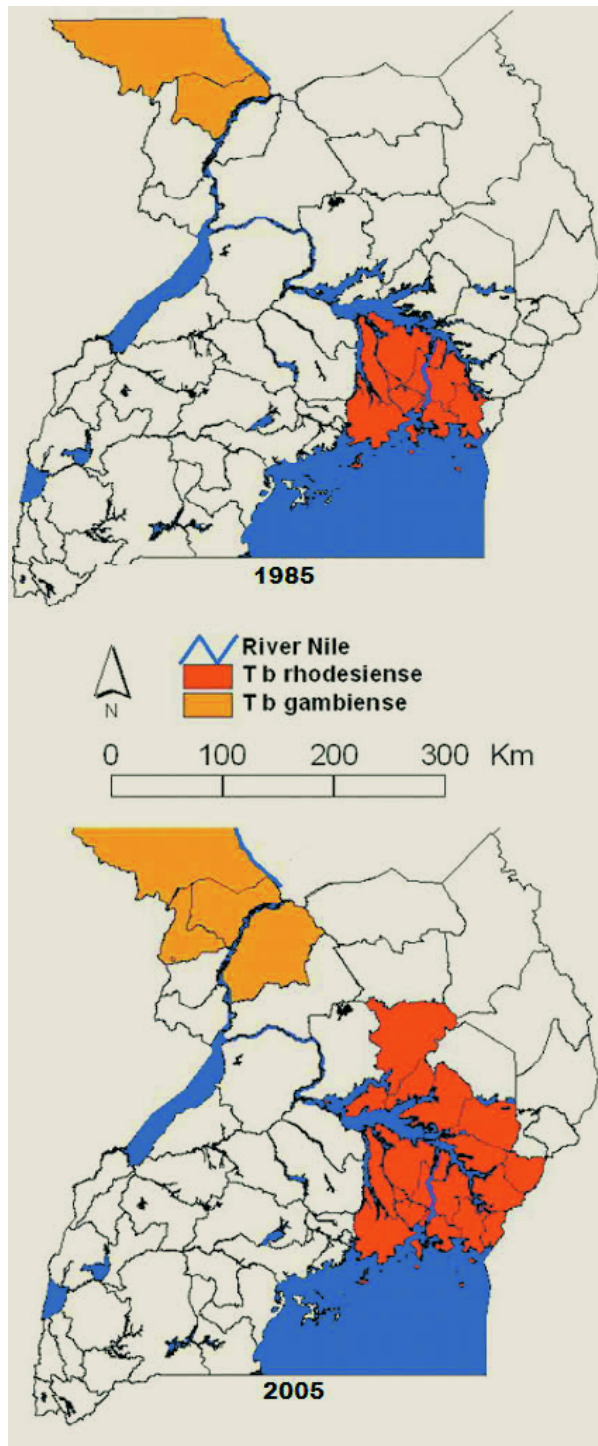


Figure 1: The conversion of Rhodesian (red) and Gambian (orange) sleeping sickness between 1985 and 2005. After Picozzi et al. (2005).

made the volunteers much more open to mass blood screening programmes.

An alternative approach to vector control has been researched in the field of molecular genetics. With the lack of a mammalian vaccine and affordable drugs making disease control difficult, Aksoy (2003) notes that the future of vector control may be the genetic disruption of the parasite transmission cycle in the invertebrate. While this requires a full understanding of the relationship between tsetse and trypanosome, the replacement of susceptible insect phenotypes with anti-pathogenic properties could result in decreased transmission.

Accompany these vastly different mitigation strategies with the application of insecticides such as deltamethrin (e.g. Torr et al., 2007a; Hargrove et al., 2012) to grazing cattle, and a focus on preventative measures which manage tsetse fly population sizes can be identified.

### Agent-based modelling

Lambin et al. (2010) review the merits of using multi-agent simulations (MAS) to model disease transmission, concluding that the technique is a good method of acquiring preliminary knowledge of a disease system, and that the representation of the dynamics of people-vector contacts in space and time are ideal to investigate scenarios that have not previously been observed or explored.

Similarly, the modelling of people-vector contacts is of particular significance given that the majority of current mitigation strategies focus on the control of tsetse fly movement and density. Lambin et al. (2010) also consider the incorporation of geographical information in epidemiological models through the use of MAS, such as the impact of land-use and land cover change. The report suggests that the benefits include an increased knowledge of transmission cycles, allowing the construction of 'pathogenic landscapes' which can subsequently provide an early warning of increased transmission risk. The benefits of incorporating geographical data into epidemiological models can be observed widely in the literature. For example, Raffy and Tran (2005) note that landscape features largely control the connectivity between hosts and vector habitats, inhibiting movement, and ultimately modifying disease risk. One of the least well integrated factors in traditional landscape epidemiology is human behaviour (Lambin et al., 2010). Despite this, different risk perception between men and women, and also by permanent and part-time residents of endemic areas, has been shown to influence the adoption of preventative measures and ultimately vary transmission risk (e.g. Stjernberg and Berglund, 2005).

We therefore saw an opportunity to build an ABM that included the disease vector (tsetse flies) and both host species (humans and cattle) interacting in a common geography so that we could assess the likely costs and benefits of widely differing management strategies for HAT.

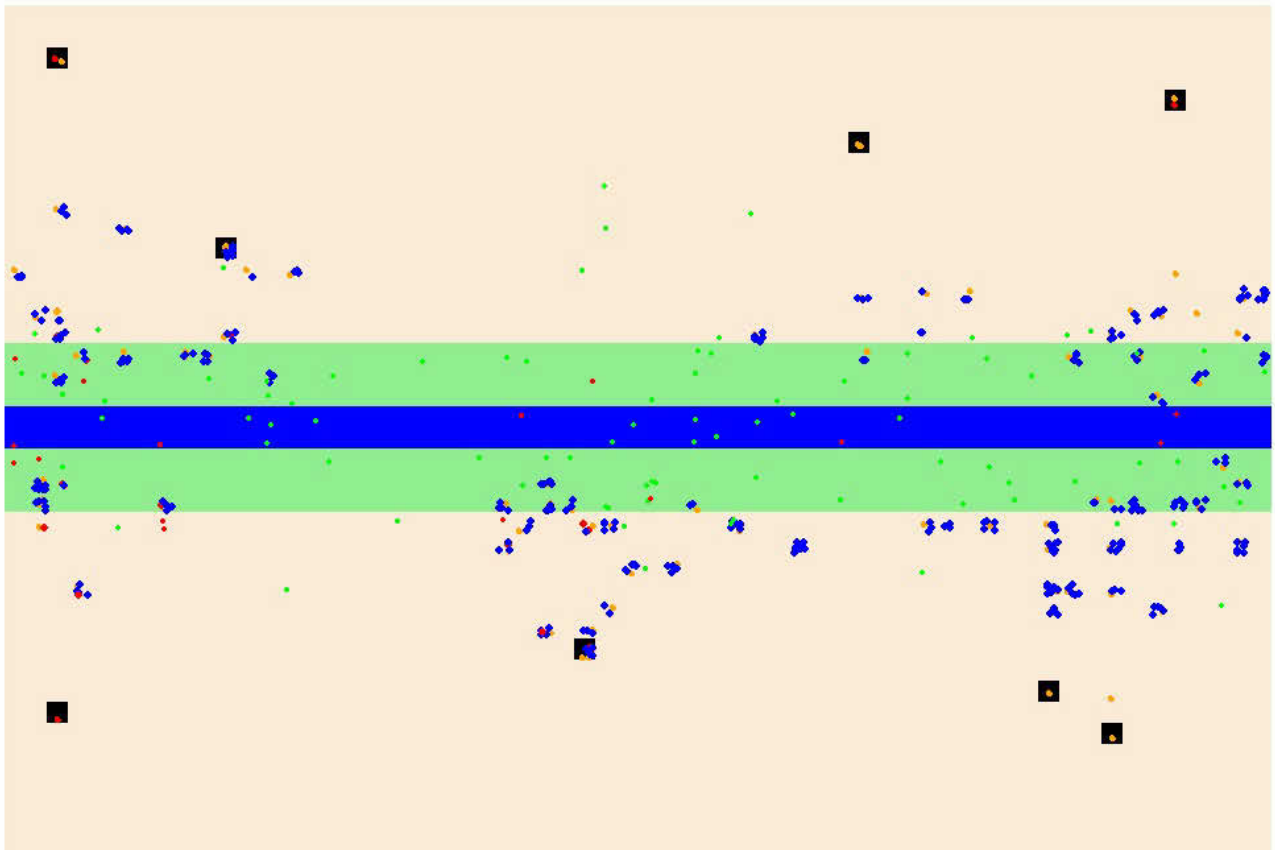


Figure 2: The simulated environment; description in text.

### The Model

The simulation incorporates an abstract spatial map and three interacting agents: tsetse flies, humans and cattle, allowing the influence of agent interaction in disease transmission to be explored. The model includes two simple daily tasks for farmer and non-farmer agents: farmers must drive their cattle to the river to drink, before grazing and returning home. Non-farmers must collect water from the river and return it to their settlement. Tsetse flies are assumed to have a natural habitat on, or near the river. Interactions between flies and the ground-based agents occur when the daily tasks take place, and humans and cattle enter the high risk, tsetse infested river area.

We begin with a simple, abstract terrain representing an area of 30 km by 20 km. The simulated land includes a river (blue) at the centre of the canvas, with flood plains either side (green), and the remaining land representing general pasture. Black icons represent small settlements or home bases for human and cattle agents (figure 2).

Within the model, time steps are grouped into the phases ‘night’ (7 pm - 5 am), ‘morning’ (5 am - 8 am), ‘day’ (8 am

- 4 pm) and ‘evening’ (4 pm - 7 pm), simulating a day with 240 time steps. These phases govern agent activity and vary exposure to the disease vector, ensuring that farmers drive their cattle to and from the river each day before grazing and returning to their settlement.

Humans are depicted in the graphical output as orange circles and are divided into farmers and non-farmers. This distinction dictates their movement pattern to and from the river. At setup, all humans are randomly assigned a settlement as a home base, with the initial cattle population randomly assigned a farmer as an owner. During the morning and day phases, all humans are required to have left their home settlements and begun either making a trip to the river to collect water (non-farmers), or driving their cattle to water before grazing (farmers). During the morning phase, there is an initial probability (10%) for each individual leaving their home per time step. When the day phase is reached, any person not to have left home yet is forced to leave.

Movement is governed using a cell desirability function, such that the rows of the map are assigned integer values which are high at the river, and decay to the edges of the



map. People move from home to the river by selecting a desirable cell from a list of their neighbouring locations. When a movement probability is exceeded for each individual, there is a 66% chance they will move to a cell with a higher value (closer to the river), and a 33% chance they will either move closer to the river or make a lateral movement, simulating different routes taken, and the subsequent spreading out of agents.

Non-farmers move directly to their homes when the river has been reached. However, if the people are farmers, a random movement element has been included to simulate the grazing of cattle. Once the cattle have been driven to the river (and whilst still in the day phase) this 'grazing' occurs with a probability per time step of going home (initially 1%). When the evening phase is reached, all remaining farmers drive their cattle to the home settlement. By the 'night' phase, all people and their cattle will have returned to their home settlement and remain there until the next morning phase, when the cycle begins again.

Should people get bitten and become infected with the disease, when they return home, they stay there, and do not resume with the next daily cycle. This simulates the debilitating nature of the disease, preventing people from undertaking their everyday duties.

As previously mentioned, cattle (blue circles) are assigned an owner at initialisation, and are programmed to follow the movement pattern of their keeper. The exception to the rule is when owners become infected and subsequently stay at home for the duration of the simulation. Under these circumstances, the cattle of the infected farmer are redistributed to uninfected people at the same home settlement, during the night phase, so that they can still be driven to the river. The redistributed cattle can be infected or uninfected and this redistribution can continue as more people become ill, until there are no longer any healthy humans in the home settlement. At this point all cattle stay at home, whether infected or not, as there are no healthy human agents to drive the cattle to water.

Tsetse flies are represented as green icons. They stay close to the river and flood plain areas which represent the natural habitat of the species. This behaviour is implemented by assigning each cell on the grid an integer value, with the central 10 rows having a uniformly high value, decaying north and south of the river so that the extreme four rows have values of 0. Tsetse flies have a lower initial movement threshold than humans to simulate faster movement. The random and directed elements of fly movement are not separated to reflect a daily routine as found with humans. Instead, flies are initially set to have a 70% chance of moving using the 'fly suitability rating' of the cells around them (i.e., moving to the neighbouring cell with the highest value). The alternative is a completely random move to any of the agent's 8 neighbouring cells. This movement regime means that, while the majority of flies will stay close to the

river and its banks, some spread is observed as stray flies can move away from their natural habitat and towards the settlements.

In this simulation, flies have a 100% chance of contracting the disease if they bite an infected cow or human, however, if the fly does not bite an infected agent for their first blood meal, the fly is re-spawned at the river to represent the removal and replacement of that agent. This action simulates the finding that flies that don't become infected on their first blood meal are much less likely to contract, and therefore, transmit the disease (e.g. Walshe et al., 2011; Aksoy, 2003). Additionally, once a fly has become infected, there is a 15-30 day delay before it becomes infective, incorporated to reflect a real life incubation period for the disease (Muller et al., 2004).

The transmission of the disease between agents is governed by a bite and infection probability, using values extracted from the literature. At the beginning of each simulation, a single fly is infected with sleeping sickness. Transmission occurs from the point of view of the tsetse fly. For every time step, each fly has a 'potential victim' list, made up of all the human and cow agents in the same cell. If this list is populated, and a randomly generated number is less than the chosen bite probability value, the fly randomly chooses one of the agents in the list as a victim.

Whether or not the bite successfully infects the human or cow agent is governed by another pair of probabilities. As humans and cows are widely considered to be differentially susceptible to *T. b. gambiense* and *T. b. rhodesiense* (e.g. Fèvre et al., 2006), different infection probabilities are incorporated for the different types of agents. The values associated with these three probabilities vary in the literature, due in part to the small sample sizes in data collection, and the complexity involved in the transmission of the disease and vector-host interaction. For example, Torr et al. (2007b) find that cattle bite rates vary with cattle age and herd size, suggesting that mean fly feeding probability can increase from 54% to 71% with an increase in herd size from 1 to 12. We take our infection probabilities from the work of Hide (1999), reporting on the dynamics of transmission of sleeping sickness during the 1988 to 1990 sleeping sickness epidemic. Hide (1999) reports a cattle infection probability of 0.0115, and a human infection probability of 0.006, based on data for the frequency of infected hosts in the population during this period in Uganda.

Consequently, the basic bite probability under investigation will be 0.54, while infection rates of 0.115 and 0.06 will be used to reflect the susceptibility of cows and humans respectively. This uses the same ratios as Hide (1999), however the values have been increased by a factor of 10 to compensate for the unrealistically small population sizes we have used (typically 100 humans, 320 cows and 100 flies). Smaller population sizes have been used as it would be computationally inefficient to attempt to simulate every

fly present in an area this size.

To simplify the simulation, and consider worst case epidemic scenarios, recovery from the disease is absent from the model. While an ideal model would incorporate some form of recovery factor, the degree of variability in diagnosis and treatment rates would be a difficult thing to include in a spatially and culturally abstract simulation.

## Results

Figure 3 shows the progression of transmission of the disease in this abstract environment with 100 humans (including 80 farmers), 320 cattle, and 100 flies over a 6 month period. 6 months was chosen as, in an environment where there is no recovery, this is how long it takes for the disease to affect the vast majority of the population, given the human infection rate of 0.06, and a cow infection rate of 0.115. Bite rate is set at an intermediate value of 0.54, thought to be significant in cattle herds (after Torr et al., 2007b).

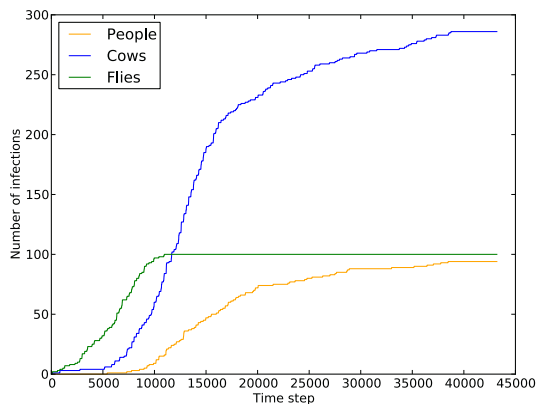


Figure 3: Progression of sleeping sickness over 6 simulated months (240 time step days); from single infected fly to complete epidemic. (Bite probability = 0.54, cow infection probability = 0.115, human infection probability = 0.06).

The six month run of the simulation shows that the fly population is the first to reach complete infection, and as suggested above, the fastest rate of cow infection occurs between months 1 (7200 time steps) and 2 (14400 time steps), once fly infection reaches approximately 50%, and sick humans start to be taken out of the equation by staying at home. The simulation shows that although fly infection occurs for 100% of the population, there are approximately 40 cows and under 10 people which are not infected after 6 months. These remaining cows are at a significantly reduced risk of infection as the simulation progresses as it is possible that all people from a certain settlement become ill, meaning that uninfected cows can be left at home with no fit human to take them to water. Even before this occurs, if it is left to

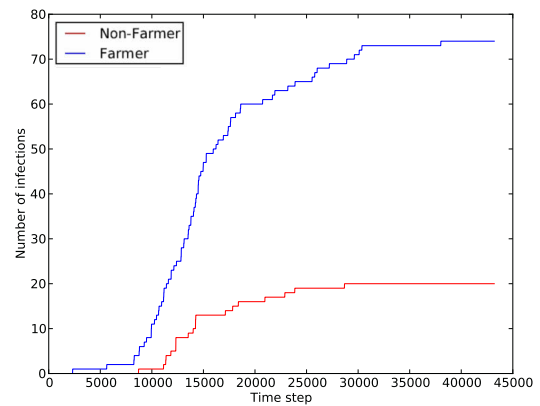


Figure 4: Infection rates of cattle driving farmers and water retrieving children (non-farmers).

one fit person to take an entire settlement's cow population to water, the spatial coverage of this herd is low, the chance of each individual being bitten is low, and a large proportion of the herd may already be infected, reducing the probability of new infections even further. Similarly, the healthy human that drives this large body of cattle is at a significantly reduced risk as the potential victim list for each fly will be more heavily populated. In this sense, even a simple, spatially abstract simulation such as this can help relate to certain theories of transmission, such as that of Torr et al. (2007b), who suggest that although bite probability increases in larger herds, there may be a degree of 'safety in numbers'.

Figure 4 shows the progression of the disease amongst the 80 farmers and 20 non-farmers in the simulation. Prior to running the simulation, one prediction may have been that the infection would propagate at a faster rate among non-farmers than farmers due to the fact that they have no cattle, and therefore when they encounter an infected fly, they are the only potential victim. As the rate of infection for each sub-class appears to be comparable for the first ten victims, this theory appears to be accurate, particularly as there are only 20 non-farmers in the population, compared to 80 farmers.

Figures 5 and 6 show the results from a series of two-month simulations where the ratio of cattle to farmers is systematically increased from 1:1 (80 farmers and cows) up to 1:7 (80 farmers, 560 cows). As before, the infection rates are a scaled representation of those found in the 1988 Ugandan sleeping sickness epidemic (Hide, 1999). These values of 0.115 for cow infection, and 0.06 for human infection, suggest the propagation of a disease which is reflective of the Rhodesian form of sleeping sickness, where cattle are the main reservoir (Batchelor et al., 2009).



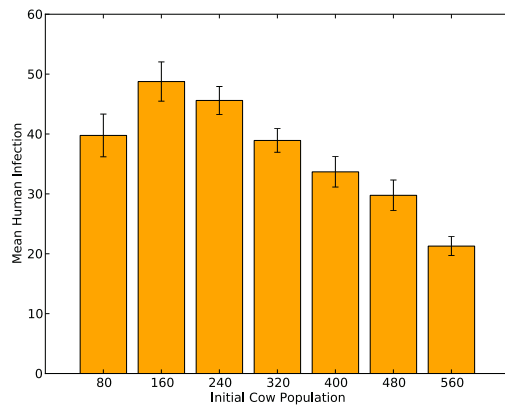


Figure 5: Mean human infection (25 repeat runs) while varying the cow population. *T. b. rhodesiense* infection rate (Bite probability = 0.54, cow infection probability = 0.115, human infection probability = 0.06).

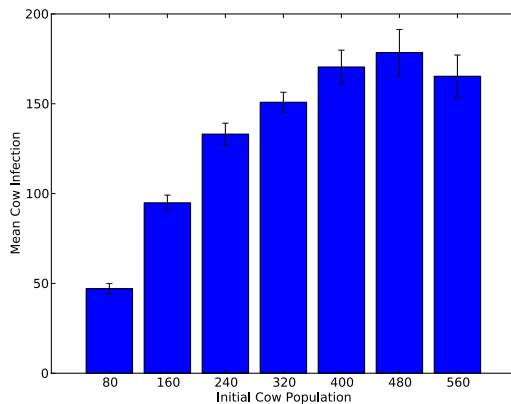


Figure 6: Mean cow infection while varying the cow population. *T. b. rhodesiense* infection rate (as figure 5).

With a 1:1 farmer to cow ratio (80 cows), overall infection is comparable between the two agent types by the end of the two month period. This is likely to be a product of the low total population, and a slower spread of the disease as a result. In a scenario where a fly meets a farmer and his cow in a grid square, there is a 50% chance of selecting either, and the bite probability is the same for both (0.54). Therefore, the only factor promoting infection in cows over humans is the infection rate, which is approximately double, but still only represents a 1 in 10 chance of infection. As a result, flies are less likely to meet infected cows or humans at the earlier stages of this two month simulation and, at this time scale, there is little promoting infection in either agent type

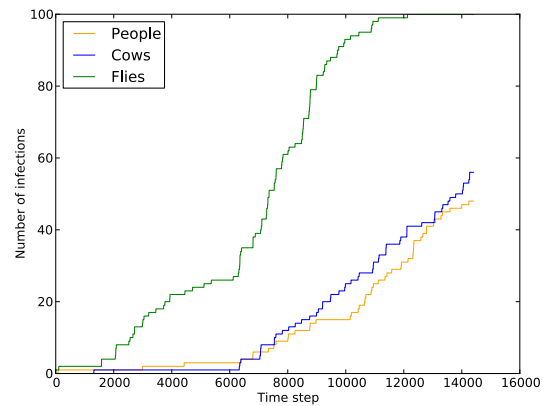


Figure 7: Example of a two month run where human and cow populations are both 80.

(see, for example, figure 7). By the end of the simulation almost all flies are infected (graph not shown) yet the relative sparsity of the human and cow populations means that disease propagation is inhibited.

When the cow population is increased to 160, mean human infection also increases, despite the greater proportion of cows to humans. This is likely to be a result of the increased rate at which the disease is transmitted, given that infection amongst the cow population significantly increases from a mean of 50 infections to a mean of 100. Although the mean increase in human infection is slight, it suggests that the idea of 'safety in numbers' does not apply for this population ratio. However, subsequent increases in the cattle population seem to create this effect, with mean human infection reaching a lower limit of 21 when the human to cow ratio is 1:7. At this point, it appears as though the number of new infections that can occur with an increasing cow population has peaked. This is likely to mean that the cow population is no longer the limiting factor in increasing infection, and instead the fly population is. Indeed, at this point the fly to cow ratio is also 1:7, and therefore a two month period may not be long enough for the vector to have a greater impact on the total population.

Figures 8 and 9 illustrate the results of a similar simulation with the infection probabilities reversed (cows = 0.06, humans = 0.115).

This scenario may be expected to produce infection data representative of the Gambian form of sleeping sickness, where humans are the primary reservoir, and cattle are affected to a lesser extent. While there are on average more infections per 100 of the human population than for the cow population, there are a few interesting points to note. Firstly, the range of mean human infection across all initial cow populations is 30-50 people. Compare this to the human infec-

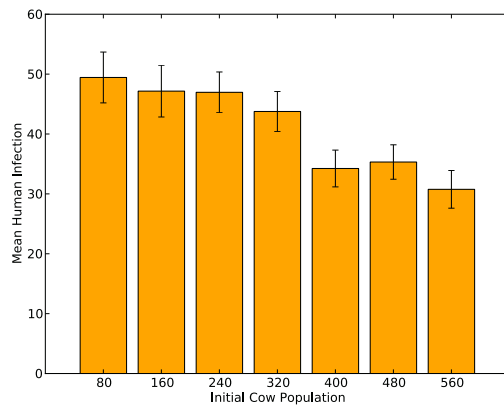


Figure 8: Mean human infection (25 repeat runs) while varying the cow population. Reversed infection rate (Bite probability = 0.54, cow infection probability = 0.06, human infection probability = 0.115).

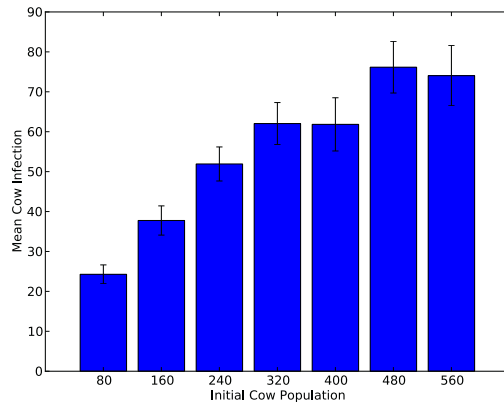


Figure 9: Mean cow infection (25 repeat runs) while varying the cow population. Reversed infection rate (as figure 8).

tion range in figure 5, which is 21-49, and changing the infection rate of humans between simulations appears to have had little effect, particularly as the shape of the plots is very similar. Although there are significantly fewer cow infections observed in figure 9 than figure 6, the data suggests that human to cattle population ratios and human infection numbers alone cannot be used to distinguish between two distinct infection rate scenarios. While only a simple simulation at this point, one can see how this may have some bearing in the real world, where knowledge of cow infection data would be very useful, but may not be widely available, or easy to collect.

## Conclusion

This report has outlined a growing problem in sub-Saharan Africa: the re-emergence of sleeping sickness at epidemic levels during the 2000s, and the risk that two distinct forms of the disease, *T.b. rhodesiense* and *T.b. gambiense*, may no longer be spatially discrete in the near future. Combine this with inaccurate diagnosis techniques and scarcely available, outdated treatment, and there are urgent reasons to investigate new means of mitigating the disease. Agent-based modelling appears to be a tool which can aid this mitigation, particularly as a significant amount of focus has been given to controlling the spread and density of the disease vector, the tsetse fly. The results from our model do not yet constitute firm predictions as there are several key parameters on which we would like to get improved data, and we have so far used a spatially abstract simulation. However, the project has conveyed the potential for the technique to incorporate a degree of spatial complexity which would be extremely difficult in a purely traditional epidemiological susceptible-infected-susceptible (SIS) model.

While the future spread of sleeping sickness appears uncertain in the affected areas, the exploration of a number of potential scenarios with agent-based modelling appears to be a sensible step in the future study of the epidemiology of the disease. Improvements that we plan to make in the second iteration of this model include a move away from an abstract gridworld in favour of GIS-derived maps of real Ugandan landscapes, allowing the identification of habitats suitable for tsetse flies, suitable watering holes for cattle, and herding routes for farmers. This incorporation of detailed maps (especially at different times of year) will allow exploration of the connectivity between host and vector habitat, which may vary seasonally due to the drying out of lakes, and the associated modification of the tsetse fly risk zone.

In addition, more attention will be given to the possibilities of ABMs for the rich representation of human daily routine and decision-making on key aspects of behaviour that can affect disease transmission. Parameters to explore may include behaviour towards the sick, the distribution of daily tasks amongst the household, the availability/successful undertaking of preventative measures, and the decision to go to a market town despite the risk of infection.

## Acknowledgements

SA was supported by the UK's Engineering and Physical Sciences Research Council via Southampton's Institute for Complex Systems Simulation and doctoral training centre. JN provided support in the construction of the simulation, the interpretation of results, and the formulation of this paper. Thanks to Prof. Peter Atkinson and Prof. Seth Bullock for their constructive feedback on the work.

## References

- Aksoy, S. (2003). Control of tsetse flies and trypanosomes using molecular genetics. *Veterinary Parasitology*, 115(2):125–145.

- Auchincloss, A. H. and Diez Roux, A. V. (2008). A new tool for epidemiology: the usefulness of dynamic-agent models in understanding place effects on health. *American journal of epidemiology*, 168(1):1–8.
- Batchelor, N. A., Atkinson, P. M., Gething, P. W., Picozzi, K., Fèvre, E. M., Kakembo, A. S. L., and Welburn, S. C. (2009). Spatial predictions of Rhodesian Human African Trypanosomiasis (sleeping sickness) prevalence in Kaberamaido and Dokolo, two newly affected districts of Uganda. *PLoS neglected tropical diseases*, 3(12):e563.
- Fèvre, E. M., Coleman, P. G., Welburn, S. C., and Maudlin, I. (2004). Reanalyzing the 1900–1920 sleeping sickness epidemic in Uganda. *Emerging infectious diseases*, 10(4):567–73.
- Fèvre, E. M., Picozzi, K., Fyfe, J., Waiswa, C., Odiit, M., Coleman, P. G., and Welburn, S. C. (2005). A burgeoning epidemic of sleeping sickness in Uganda. *Lancet*, 366(9487):745–7.
- Fèvre, E. M., Picozzi, K., Jannin, J., Welburn, S. C., and Maudlin, I. (2006). Human African trypanosomiasis: Epidemiology and control. *Advances in parasitology*, 61:167–221.
- Fèvre, E. M., Wissmann, B. V., Welburn, S. C., and Lutumba, P. (2008). The burden of human African trypanosomiasis. *PLoS neglected tropical diseases*, 2(12):e333.
- Gumerman, G. J., Swedlund, A. C., Dean, J. S., and Epstein, J. M. (2003). The evolution of social behavior in the prehistoric American southwest. *Artificial life*, 9(4):435–444.
- Hargrove, J. W., Ouifki, R., Kajunguri, D., Vale, G. A., and Torr, S. J. (2012). Modeling the control of trypanosomiasis using trypanocides or insecticide-treated livestock. *PLoS neglected tropical diseases*, 6(5):e1615.
- Hide, G. (1999). History of sleeping sickness in East Africa. *Clinical microbiology reviews*, 12(1):112–25.
- Hotez, P. J. and Kamath, A. (2009). Neglected tropical diseases in sub-saharan Africa: review of their prevalence, distribution, and disease burden. *PLoS neglected tropical diseases*, 3(8):e412.
- Joja, L. L. and Okoli, U. A. (2001). Trapping the vector: community action to curb sleeping sickness in southern Sudan. *American journal of public health*, 91(10):1583–5.
- Lambin, E. F., Tran, A., Vanwambeke, S. O., Linard, C., and Soti, V. (2010). Pathogenic landscapes: interactions between land, people, disease vectors, and their animal hosts. *International journal of health geographics*, 9:54.
- Lansing, J. S. and Kremer, J. N. (1993). Emergent Properties of Balinese Water Temple Networks: Coadaptation on a Rugged Fitness Landscape. *American Anthropologist*, 95(1):97–114.
- Matovu, F. S. (1982). Rhodesian sleeping sickness in South-Eastern Uganda: (the present problems). *East African medical journal*, 59(6):390–3.
- Mbulamberi, D. B. (1989). A review of human African trypanosomiasis (HAT) in Uganda. *East African medical journal*, 66(11):743–7.
- Muller, G., Grébaud, P., and Gouteux, J.-P. (2004). An agent-based model of sleeping sickness: simulation trials of a forest focus in southern Cameroon. *Comptes Rendus Biologies*, 327(1):1–11.
- Okiria, R. (1985). The prevalence of human trypanosomiasis in Uganda, 1970 to 1983. *East African medical journal*, 62(11):813–816.
- Picozzi, K., Fèvre, E. M., Odiit, M., Carrington, M., Eisler, M. C., Maudlin, I., and Welburn, S. C. (2005). Sleeping sickness in Uganda: a thin line between two fatal diseases. *BMJ (Clinical research ed.)*, 331(7527):1238–41.
- Raffy, M. and Tran, A. (2005). On the dynamics of flying insects populations controlled by large scale information. *Theoretical population biology*, 68(2):91–104.
- Simarro, P. P., Cecchi, G., Paone, M., Franco, J. R., Diarra, A., Ruiz, J. A., Fèvre, E. M., Courtin, F., Mattioli, R. C., and Jannin, J. G. (2010). The Atlas of human African trypanosomiasis: a contribution to global mapping of neglected tropical diseases. *International journal of health geographics*, 9(1):57.
- Stjernberg, L. and Berglund, J. (2005). Tick prevention in a population living in a highly endemic area. *Scandinavian Journal of Public Health*, 33(6):7.
- Torr, S. J., Maudlin, I., and Vale, G. A. (2007a). Less is more: restricted application of insecticide to cattle to improve the cost and efficacy of tsetse control. *Medical and Veterinary Entomology*, 21(1):53–64.
- Torr, S. J., Prior, A., Wilson, P. J., and Schofield, S. (2007b). Is there safety in numbers? The effect of cattle herding on biting risk from tsetse flies. *Medical and veterinary entomology*, 21(4):301–11.
- Walshe, D. P., Lehane, M. J., and Haines, L. R. (2011). Post eclosion age predicts the prevalence of midgut trypanosome infections in Glossina. *PloS one*, 6(11):e26984.
- Watkins, A., Noble, J., and Doncaster, C. P. (2011). An agent-based model of jaguar movement through conservation corridors. In Lenaerts, T., Giacobini, M., Bersini, H., Bourguine, P., Dorigo, M., and Doursat, R., editors, *Advances in Artificial Life, ECAL 2011: Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems.*, pages 846–853. MIT Press.

# Cell Division Behaviour in a Heterogeneous Swarm Environment

Adam Erskine, J. Michael Herrmann

The University of Edinburgh, School of Informatics, IPAB, 10 Crichton Street, Edinburgh EH8 9AB, U.K.  
a.erskine@sms.ed.ac.uk

## Abstract

We present a system of virtual particles that interact using simple kinetic rules. It is known that heterogeneous mixtures of particles are producing particularly interesting behaviours. Here we present a two-species swarm in which a behaviour emerges that resembles cell division. We show that the dividing behaviour exists across a narrow but finite band of parameters and for a wide range of population sizes. In a two dimensional environment the swarm's characteristics and dynamism manifests differently from those observable in a three dimensional environment. In further experiments we show that repeated divisions can occur if the system is extended by a biased equilibrium process to control the split of populations. We propose this repeated division behaviour provides a simple model for cell division mechanisms, which relates to discussions of the origin of life and is of interest for the formation of morphological structure and to swarm robotics.

## Introduction

We investigate emergent behaviours found arising from the interactions within a heterogeneous swarm. The interactions are in the manner of that originally described by Craig Reynolds (Reynolds, 1987). He introduced a simple algorithm showing that such a swarm could manifest flocking behaviours. Each particle is influenced only by other particles in its local neighbourhood. Each update of the model represents a discrete time step. On each update every particle is drawn toward the centre of mass of its neighbours, aligns its velocity with its neighbours and is pushed away from any particles too close. Reynold's swarms were homogeneous.

Sayama (2009) extended this approach allowing multiple swarms to interact. Each swarm may have different sets of parameters. A set of parameters may be thought of as defining a species. By mixing two or more species of swarms unusual structures and dynamic behaviours have been seen (Sayama, 2010, 2012b,a). Many swarms could be identified that have a distinct biological look to them: cells, amoebas, diatoms abound. It is tempting to see the dynamics of the so-called *swarm chemistry* as a simple model for the real life counterparts of these forms.

We extend the heterogeneous swarm algorithm to include both growth and biased equilibrium mechanisms. Our explorations have found a set of species that show cell division like behaviour. Density and entropy measures allow us to make broad categorizations of behaviours. Single homogeneous swarm show limited behaviours, but more complex emergent behaviours are apparent with just two interacting species. Our investigations explore the robustness of this behaviour under parametric variation. Specifically we studied:

- How cell division is affected by the total size of the swarm and the populations of each subspecies.
- The differences in the behaviour exhibited in 2D and 3D environments.
- How cell division is affected by variation of several of each swarm's defining parameters.

Structure and form abound in and between biological organisms. Much of this comes about via self organization. One benefit of this is that its resultant emergent forms are, in some sense, available for free. Structure emerges from interactions without the need for it to be explicitly coded. An understanding of these rules and their application allow us the possibility of reusing this free structure in robotic systems. Self-organization of structures, self repair or growth without explicit command and control is beneficial. This approach may provide a model that allow us look at the automatic creation of morphological artefacts and dynamic behaviours. The tendency of many swarms to mirror biological forms, albeit superficially, raises the question of whether they can also be a model of biological processes.

Theories on the origin of life often invoke mechanisms to assure that proto-replicators are held in close association: within rock fissures; agglomeration at thermal vents; within the wind blown organic foams formed in the sea. Self-organized structures offer options for such discussions. A similar argument is made (Hutton, 2002) with reference to artificial chemistry. However this model is limited to the organizational dynamics arising from its kinetic interactions.

Single cell division and the dynamics of small multicellular groups contain the ebb and flow of chemical gradients, protein interactions and gene expressions. Whilst much is known, the precise chemo-mechanical details are still there for investigation. We propose that the dynamics of our cell division swarms may offer a simple model that allows some of these investigations. In order to allow this we require that a robust repeating cell division like mechanism be implemented. Thus we also look at modifications made to enable the observed cell division behaviour to repeat.

## Background

D'arcy Thompson detailed many roles that physical processes might play in the morphological development of creatures and their artefacts (Thompson, 1917). He saw that the forms that soap bubbles took as their surface energies pulled and found equilibria bore resemblances to biological forms. He believed that this was not mere coincidence. It has been shown that this idea is indeed true — at least in part. Honeycomb, its hexagonal packing and shape of end caps, are both found in bubble foams but are not derived from a bubble formation mechanism (Ball, 2011). However the packing of the four cones in the ommatidia of a fly's compound eye may be due a mechanism of simple squeezing together like bubbles. Ball also documents work that notes that the spicule structures of sponges appears to form via a mechanism whereby a bubble array is created and then inorganic compounds are allowed to permeate the interstices of the bubble matrix. The creature is leveraging the free structure from what Ball refers to as a fossilized foam. The processes at all scales of life are complex when compared to the simple mechanisms that our model uses. And yet simple processes may shed light on the forms that life can take. Finite subdivision rules have been used to model cell division previously.

Reynolds' flocking algorithm have been subject to numerous variations, adding in: assumed fear, or leadership roles, or desire to stay close to roost sites etc. It has been shown (Feder, 2007) that in starlings it is the number of neighbours (not radius), that is important, and that the influence of neighbours was spatially anisotropic. Nearest neighbour interactions combined with an energy minimization argument has been used to generate line and vee formation flocks (Klotsman and Tal, 2011). These homogeneous swarm algorithms have been further extended by combining multiple 'species'. We should mention here again the studies of Sayama in particular on the relationship between 2D and 3D species (Sayama, 2012b). An evolutionary approach was adopted to discover interesting heterogeneous swarms (Sayama, 2010, 2012a).

Local interactions in biology have been much studied. Quorum sensing, the switching of behaviours due to local sensing, is seen in a large range of organisms from bacteria to honeybees (Miller and Bassler, 2001; Seeley

et al., 2006). Various insects employ local microrules to drive artefact construction (Camazine et al., 2001). It has been shown (Schmickl and Hamann, 2011; Kengyel et al., 2009; Bodi et al., 2009) that bees, through local interactions, locate areas of a target temperature. Such biological inspirations have informed swarm robotic work. Review documents (Bayindir and Sahin, 2007; Mohan and Ponnambalam, 2009) highlight the extensive range of behaviours that may be implemented from swarm robotic interactions, including: pattern formation; aggregation; chain formation; self-assembly; coordinated movement; hole avoidance; foraging; self-deployment; grasping; pushing; caging.

## Method

The basic heterogeneous swarm algorithm (Sayama, 2012b) gives each particle a set of parameters. Each particle's update of position and velocity is influenced only by its local particles within a specific neighbourhood radius. Each particle has a preferred normal speed, the maximum speed being bounded. Parameters  $c_1$ ,  $c_2$  and  $c_3$  scale the influence of the neighbouring particles. The  $c_1$  parameter is a measure of cohesion, the strength of pull toward the mean neighbour position. The  $c_2$  parameter is a measure of alignment, the strength of pull toward mean neighbour velocity. The  $c_3$  parameter is a measure of avoidance, the strength of push from close neighbours. On each update of the swarm each particle uses neighbouring particles to update its position and velocity.

$\mathcal{N}$  is the set of particles centred on particle  $i$  and being within particle  $i$ 's neighbourhood radius. The average position of these is

$$\langle \mathbf{x} \rangle = \frac{1}{|\mathcal{N}|} \sum_{j \in \mathcal{N}} \mathbf{x}_j.$$

The average velocity of the particles within the neighbourhood radius of particle  $i$  is

$$\langle \mathbf{v} \rangle = \frac{1}{|\mathcal{N}|} \sum_{j \in \mathcal{N}} \mathbf{v}_j.$$

The acceleration of particle  $i$  is given by

$$\begin{aligned} \mathbf{a}_i = & -c_1 (\mathbf{x}_i - \langle \mathbf{x} \rangle) - c_2 (\mathbf{v}_i - \langle \mathbf{v} \rangle) \\ & + c_3 \left( \sum_{j \in \mathcal{N}} (\mathbf{x}_i - \mathbf{x}_j) / |\mathbf{x}_i - \mathbf{x}_j|^2 \right). \end{aligned}$$

The dynamics are further modified by the  $c_4$  parameter which is a probability of ignoring the neighbours' effects. The particle's velocity is updated using the acceleration  $\mathbf{a}_i$ .

$$\mathbf{v}'_i \leftarrow \mathbf{v}_i + \mathbf{a}_i.$$



The magnitude of a particle's velocity has an upper bound. This is one of the swarm's parameters. Similarly each swarm has a parameter that is the preferred magnitude of the particles velocity. If a particle is not travelling at this preferred velocity,  $\mathbf{v}_n$ , then parameter  $c_5$  is then used to nudge the velocity back to its toward its preferred velocity using

$$\mathbf{v}_i \leftarrow c_5 (\mathbf{v}_n / |\mathbf{v}_i'| \cdot \mathbf{v}_i') + (1 - c_5) \mathbf{v}_i'.$$

Finally each particle's position is updated using

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i.$$

## Quantification

The eight parameters ( $c_1$  through  $c_5$ , neighbourhood radius, speed and maximum speed) define a large parameter space. To search this space we require automated means to detect behaviours of interest.

In our swarms we can calculate the average density of particles. This density measure differentiates single blobs from both dispersed swarms and multiple blobs: single blobs show a higher density. We note that this may not always be true: a large hollow single blob may be less dense than multiple blobs that are close together. A second measure, a spatial entropy, allowed differentiation between multiple blobs and dispersed swarms. It has been suggested (Bonabeau et al., 1999) that a spatial entropy can be defined as

$$H = - \sum_k P(k) \log P(k),$$

where  $P(k)$  is the fraction of particles found in patch  $k$ .  $H$  decreases as clusters form. We used patches that are always cubes of side 0.1 times the maximum extent of the swarm i.e. the minimal cube containing the swarm is split into 1000 patches. Two similar treatments are made in (Batty, 1974) and (Wolfram, 1984).

We also use the Kullback-Leibler divergence from an evenly distributed population as a measure. This is defined by

$$D_{KL} = \sum_k P(k) \log \left( \frac{P(k)}{Q(k)} \right),$$

where  $P$  is the distribution of the particle positions and  $Q$  is the distribution of an evenly dispersed swarm. Note that since  $Q$  is evenly distributed, we have simply  $D_{KL} = \log \left( \frac{1}{Q(x)} \right) - H$ . For the cell division like behaviour  $D_{KL}$  thus increases when the swarm has divided into separate clumps.

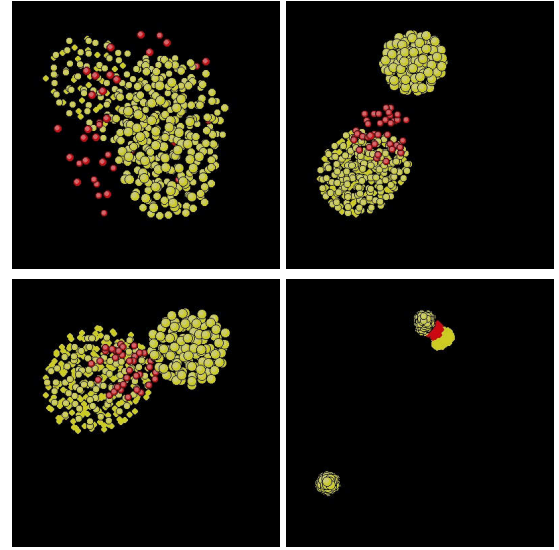


Fig. 1: Typical evolution of cell division in our swarm. Top left shows red particles as a toroid about the yellows. Top right shows the yellow swarm divided in two with a separate red swarm. Bottom left has the reds rejoining the larger yellow blob. Finally bottom right, the process repeating and is shown at a larger scale.

## Results

### Single species characterization

A homogeneous swarm appears to exhibit behaviours drawn from a fairly limited palette of possible behaviours. We note four behaviours: full dispersal, blob or sphere, multiple blobs, and one we call a point swarm (all particles collapse toward a single point). In full dispersal the particles separate and move apart, there is little or no tendency to aggregate. In a blob the particles form a sphere (or approximate sphere) or shell of a sphere. Multiple blobs are simply a multiple version of the last form. Point swarms are seen for swarm parameters where the avoidance value is at or near zero. This results in all particles collapsing to a single point. This state tends to not show as a sphere or a point. Instead the particles, which all exist in a tiny spatial volume, show as an irregular clump of particles that jump about. Particles have discretised speeds so at each update a particle tends towards the average position of the clump, but the step size is larger than the size of the clump, thus the particles are unable to actually occupy a single point.

We find that the four single species states can be classified by the density and spatial entropy (or Kullback-Leibler divergence). By sweeping through the parameter space of the cohesion and avoidance parameters it was possible to find regions of each of the four swarm types. The spatial entropy and densities were measured for each. A visual check of the final state of the swarm was also made. The



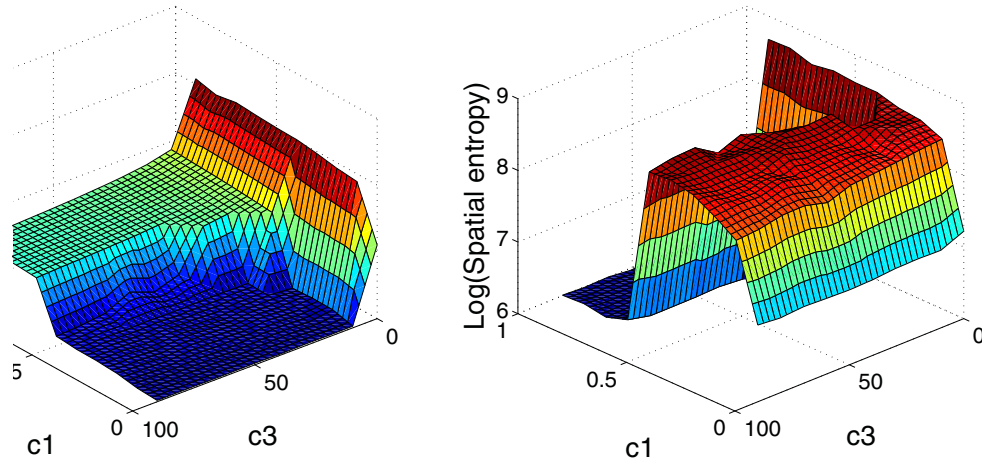


Fig. 2: Density and entropy measurements for a homogeneous swarm as a function of its cohesion ( $c_1$ ) and avoidance ( $c_3$ ) parameters. The logarithm of each measure is plotted in order to squash the vertical extent of the surface plots as the range of values extends over several decades.

measures were plotted against the parameters to generate the surface plots shown in Fig. 2.

### Cell division behaviour species

We present two species of swarms that individually formed single blobs (multiple blobs if their populations were large enough), but in combination result in a cell division like behaviour. Typical stages of this are shown in Fig. 1. The values for the parameters used in these swarms are shown in Tab. 1.

spc	rad	spd	msh	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$
1	20.5	1.94	20.7	1	1	18.6	0.05	1
2	300	15.58	37.08	1	0.05	9.11	0.47	0.61

Tab. 1: Parameter values for the cell division like behaviour swarms. Headings are: spc = swarm species, rad = neighbourhood radius, spd = normal speed, msh = maximum speed,  $c_1$  = cohesion,  $c_2$  = alignment,  $c_3$  = avoidance,  $c_4$  = whim,  $c_5$  = speed control.

When displaying the particles the parameters  $c_1$  through  $c_3$  are used to define the displayed colour of the particles. Here, species 1 displays as a yellow colour and species 2 as a red colour. Clearly if we alter these parameter values the colours will alter. For descriptive convenience we choose to describe the two swarms as the yellow and the red swarms respectively. Typically the swarm population constructed of around ten yellow particles for every red particle. When this heterogeneous swarm is run the initially mixed species separate. A toroid of red particles forms about the yellows until a split occurs. The separate yellow blobs move apart, with the red particles forming a blob in between. At some

point the reds are drawn into one of the yellow blobs and the process repeats. The repetition only occurs within a single blob of the yellows.

### Comparison with 2D Swarm Chemistry

We explored the differences in 2D and 3D behaviour of our swarms. With no change to the swarm parameters cell division behaviour still occurred. Differences in the 2D version included: the red particles travel to the inside of a yellow circle of particles causing an inside out division to occur; the separated blobs do not travel apart; and the red particles do not get drawn back into one of the yellow blobs.

An outside in division was achieved via modification of both swarms' parameters, Fig. 3. As parameters have been changed, the particles no longer appear as red and yellow but as magenta and cyan respectively. Now the red particles form a ring around the yellow circle and squeeze it until division occurs. Again the separate parts do not travel apart.

It is possible that reintegration of the red particles with one of the yellow blobs would occur if the swarm was left to run. It is also possible that with further parameter modification a recipe may be found that results in the split parts separating.

### Robustness under population dynamics

**Yellow versus red populations.** We varied the two species' populations to determine the limits on the cell division like behaviour. Each run lasted for 2000 time ticks. The density and entropy measures were captured at the end of each run. For confirmation the final state of the swarm was captured as an image. Yellow populations were varied over a range from 100 to 550 in steps of 50, and red population over the range 10 to 90 in steps of 10. Fig. 4

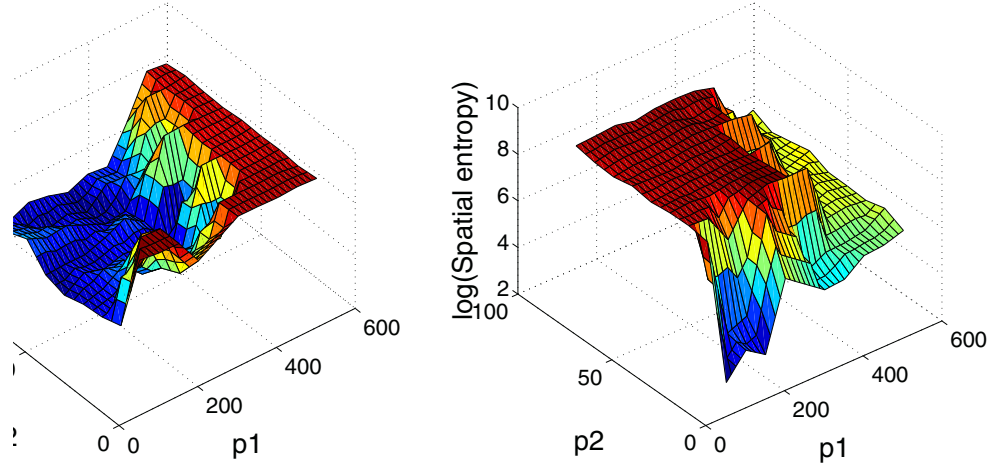


Fig. 4: Density and entropy measurements for a heterogeneous swarm as a function of its yellow and red populations ( $p_1$  and  $p_2$ ). The logarithm of each measure is plotted in order to squash the vertical extent of the surface plots as the range of values extends over several decades.

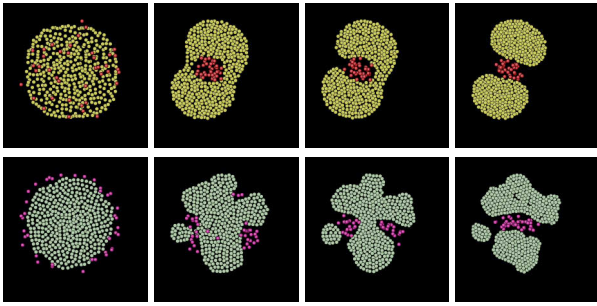


Fig. 3: 2D cell division. Upper row shows an ‘inside out’ division. Lower row shows an ‘outside in’ division.

shows the density and entropy measures as a surface plot for all combinations of these populations. Cell division is marked by low density (blue on left hand plot) and high entropy (red on right hand plot). We see that the cell division behaviour extends over a wide range of populations. Very low red or high yellow populations tend to never show cell division. The line between division and no division is noisy. We assume this is due to variability in starting position of particles and/or the arbitrary duration of each run. We explore both of these possibilities.

We fixed the red population at 50, and executed 5 runs for yellow populations varying from 300 to 600 in steps of 25. When the yellow population is below 375 division always occurred. For populations above 450 it never occurred. In the range between division may or may not occur. The difference between each run was the randomized initial positions of the particles in the swarms. The KL divergence and the density (averaged over the 5 runs) are summarized

in Fig. 5. The density increases and the KL measure drops above a population of 350 coinciding with the onset of swarms that fail to divide. When division never occurs the values level off.

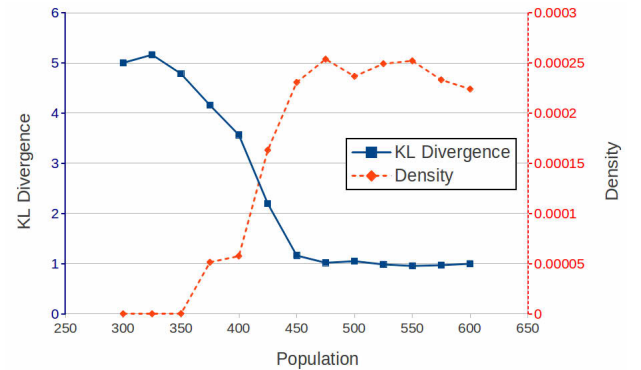


Fig. 5: Density and Kullback-Leibler divergence measures as function of yellow population after 2000 time ticks. For a fixed population of red particles (50), we vary the population of the yellow swarm (from 300 to 600).

**Effect of lengthening run time.** We repeated the previous investigation but allowing the model to run now for 10000 steps. There is still no distinct population boundary between split/no split behaviour. Yellow populations less than 425 always result in division. Those greater than 475 never divide. Populations between these limits may divide. Fig. 6 confirms this observation in that the step up in density occurs at higher yellow populations. Executing the swarm for still

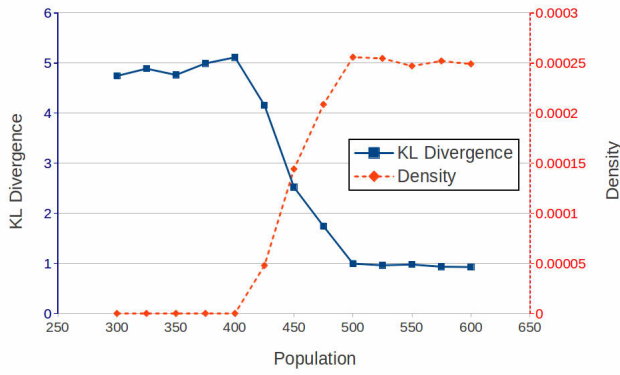


Fig. 6: Density and Kullback-Leibler divergence measures as function of yellow population after 10000 time ticks. For a fixed population of red particles (50), we vary the population of the yellow swarm (from 300 to 600).

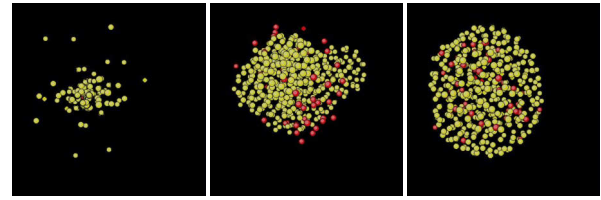
longer durations suggested that with relatively small red swarms the whole swarm may be unable to divide. However when the red population was increased (to 180) the swarm which appeared to be stable would occasionally eject a small blob of yellow particles. This suggests that such a swarm may slowly lose yellow particles until the remaining yellow blob is small enough to show the normal division behaviour.

### Robustness under parameter variation

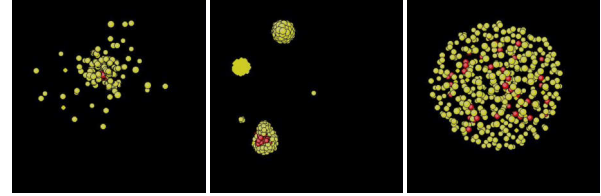
A full search of the parameter space is currently too onerous. Therefore we choose a simpler approach. We look to vary single parameters whilst keeping all other parameters unchanged. We vary the parameter being studied until the cell division behaviour disappears.

**Variation of neighbourhood radius.** Using a yellow:red population mix of 300:50 we varied, independently, the neighbourhood radii of each swarm. For cell division behaviour the red species was required to have a neighbourhood radius greater than 125 and for the yellow ‘species’ it needed to be within the range of about 13 to 25. Samples are shown in Fig. 7. Each swarm was run for 2000 time ticks. Yellow radii above 28 result either in a single cloud or have the red particles held within the yellows.

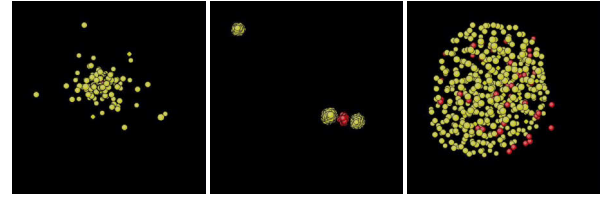
**Variation of avoidance and cohesion parameters.** We separately swept through combinations of avoidance and cohesion parameters. First we varied red avoidance between 5 and 40, yellow between 10 and 60. Then we varied the red and yellow cohesion values from 0.2 to 1.0. A number of different behaviours were noted. Several behaviours would not be distinguishable via the use of measurements alone, so each run was watched and categorized. A single run of each permutation was made. All runs lasted 2000 time ticks. Tab. 2 shows the results for avoidance variation and Tab. 3



(a) Red radius = 100, yellow radii of 10, 20, 30.



(b) Red radius = 125, yellow radii of 10, 20, 30.



(c) Red radius = 300, yellow radii of 10, 20, 30.

Fig. 7: Final states of each run. Examples of neighbourhood radius variation.

shows the results for cohesion variation.

Cell division behaviours exist over narrow ranges of both these parameters. Cell division behaviour of the sort we have been looking at is thus very sensitive to the values of both avoidance or cohesion parameters. As with the other parameter studies whether this is true for other population and parameter mixes is unknown. It appears that for small yellow avoidance values ( $c_3 \leq 40$ ) the red avoidance value needs to be around half that of the yellow value for any division to occur. Given the parameter set of the swarms, it appears that larger yellow cohesion values are needed to stop the yellow swarm from disintegrating. Perhaps above this level (around 0.6) the yellow swarm requires a greater ‘pull’ from the reds to begin to divide. As with the other parameter studies whether this is true for other population and parameter mixes is unknown.

### Repeated division

The cell division behaviour in the previous sections splits a clump of yellow particles in two. Only one of those clumps will subsequently divide again. This occurs as the red particles tend to only associate with the larger clump of yellow particles. In order for this division behaviour to be seen as a possible model for real world division we needed a mechanism that would allow any yellow clump to potentially divide. In (Sayama, 2012a) each particle is modelled as expressing one parameter set drawn from a group of parameter sets. This formulation allowed a

Avoidance values	Yellow=10	Yellow=20	Yellow=30	Yellow=40	Yellow=50	Yellow=60
Red=5	3D	2D	2D	2D	2D	Y
Red=10	0	3D	2D	2D	2D	2D
Red=15	0	0	0	3D	3D	2D
Red=20	0	0	0	0	3D	2D
Red=30	0	0	0	0	3D	Y
Red=40	0	0	0	0	3D	Y

Tab. 2: Division types as function of avoidance parameter,  $c_3$ , for a selection of the parameter variations tried. Categories are: ‘0’ — No division seen, reds may form toroid round yellows. ‘3D’ — Division seen, behaviour was characteristic of the standard 3D cell division. ‘2D’ — Considered the same as 2D case. Inside out split but clumps are largely static after split. Reds may be drawn in. ‘Y’ — Yellows disintegrate into small clumps, reds form their own clump.

Cohesion values	Yellow=0.2	Yellow=0.4	Yellow=0.6	Yellow=0.8	Yellow=1.0
Red=0.2	Y	Y	Y	0	0
Red=0.4	Y	Y	3D	0	0
Red=0.6	Y	Y	0	0	0
Red=0.8	Y	Y	3D	0	0
Red=1.0	Y	2D	2D	3D	3D

Tab. 3: Division types as function of cohesion parameter,  $c_1$ . Categories are as per Tab. 2.

natural extension to evolutionary techniques to be applied. We choose a similar approach. Each particle expresses itself either as a red or a yellow particle. There is a small probability that any particle may change the behaviour it expresses. This is modelled as a biased equilibrium processes. Each yellow, on being chosen to pick a behaviour, will select changing to red with a 0.1 probability. Each red will select changing to yellow with a 0.9 probability. This ensures a rough 90:10 percent mix in the population, but allows any clump of yellow particles to develop a red population. This mechanism only works as a divided cell tends to move apart. If the parts remain close, either by artificial confinement or as would be the case in the 2D version, then any new reds in one clump tend to be immediately sucked into the clump with the larger red population.

This mechanism alone provides for each clump to continue to divide over time. However, as clumps do not

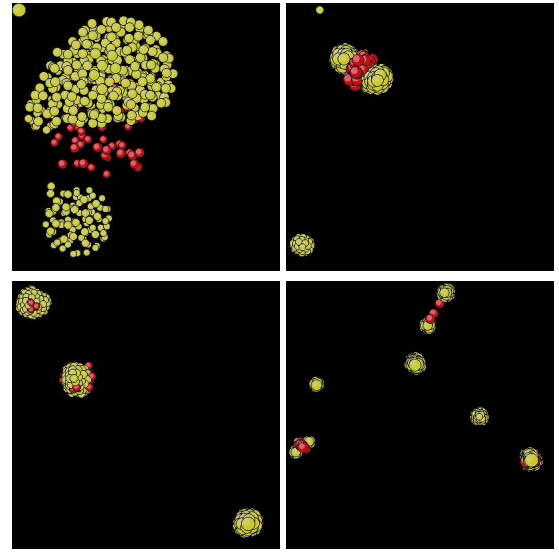


Fig. 8: Repeated division. Top left shows first division. Top right shows the second division. Bottom Left shows multiple blobs with red particles from the bias equilibrium process. Bottom right shows multiple divisions occurring.

tend to recombine the ultimate future for this approach is a dispersed swarm. We added a growth mechanism to allow clumps to increase in size. New particles would be created close to randomly chosen existing particles. This can be viewed as new particles being recruited from the environment. Fig. 8 shows some examples from a swarm that implements both the biased equilibrium and growth mechanisms. The swarm still tends to appear somewhat dispersed, however, there are still many clumps that continue to divide.

## Discussion

We presented a heterogeneous swarm that exhibits cell division like behaviour. Prior to dividing, the red particles form a toroid — but only because the yellows support it. There are configurations where this appears a long lived phenomenon. Division occurs for a wide range of swarm sizes, but there appears to be a size above which the yellow swarm tends to stability. We found some evidence that such a swarm may gradually lose yellow particles suggesting that cell division may reappear if the swarm runs for long enough. Balancing the growth and biased equilibrium can be hard and the population will tend to fragment. It would be appealing to improve the linkage between these mechanisms so that division would become more regularly periodic.

We observed differences in the emergent behaviour depending whether the swarm ran in a 2D or 3D environment. If the parameter values used in a 3D environment were used, unchanged, in a 2D environment



then we observed an ‘inside out’ division. This resulted in a relatively static set of divided clumps. By modifying the parameter values used we were able to recapture the ‘outside in’ division seen in 3D. This still failed to show the full dynamics seen in 3D. However, the fact that there are parameter mixes that show behaviour in 3D that matches that seen in 2D suggest the opposite may also be true.

The cell division behaviour was sensitive to the swarms’ parameter recipes. Yellow neighbourhood radius needs to be in a narrow band. The red neighbourhood radius appears to have a lower limit, while much larger values seem to result in division behaviour. Cell division behaviour is seen only across a narrow band of both avoidance and cohesion parameters. On one side of the band no division is observed. On the other side either an ‘inside out’ division similar to that seen in 2D, or a spontaneous yellow disintegration that requires no interaction with the red particles, is observed.

The inclusion of a biased population equilibrium and growth mechanisms enabled the swarm to show ongoing cell division like behaviours.

### Acknowledgements

Adam Erskine’s Ph.D. is funded by the Engineering and Physical Sciences Research Council (EPSRC).

### References

- Ball, P. (2011). *Shapes: Nature’s patterns: A tapestry in three parts*, volume 1. Oxford University Press.
- Batty, M. (1974). Spatial entropy. *Geographical Analysis*, 6(1):1–31.
- Bayindir, L. and Sahin, E. (2007). A review of studies in swarm robotics. *Turkish Journal of Electrical Engineering and Computer Sciences*, 15(2):115–147.
- Bodi, M., Thenius, R., Schmickl, T., and Crailsheim, K. (2009). Robustness of two interacting robot swarms using the beeclust algorithm. In *MATHMOD 2009 - 6th Vienna International Conference on Mathematical Modelling*.
- Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems*. Oxford University Press.
- Camazine, S., Franks, N. R., Sneyd, J., Bonabeau, E., Deneubourg, J.-L., and Theraulaz, G. (2001). *Self-Organization in Biological Systems*. Princeton University Press.
- Feder, T. (2007). Statistical physics is for the birds. *Physics Today*, 60(10):28–30.
- Hutton, T. J. (2002). Evolvable self-replicating molecules in an artificial chemistry. *Artificial Life*, 8(4):341–356.
- Kengyel, D., Schmickl, T., Hamann, H., Thenius, R., and Crailsheim, K. (2009). Embodiment of honeybee’s thermotaxis in a mobile robot swarm. In *Proceedings of the 10th European conference on Advances in artificial life: Darwin meets von Neumann-Volume Part II*, pages 69–76. Springer-Verlag.
- Klotsman, M. and Tal, A. (2011). Animation of flocks flying in line formations. *Artificial Life*, 18(1):91–105.
- Miller, M. and Bassler, B. (2001). Quorum sensing in bacteria. *Annual Reviews in Microbiology*, 55(1):165–199.
- Mohan, Y. and Ponnambalam, S. (2009). An extensive review of research in swarm robotics. In *Nature & Biologically Inspired Computing*, pages 140–145. IEEE.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics*, volume 21, pages 25–34. ACM.
- Sayama, H. (2009). Swarm chemistry. *Artificial Life*, 15(1):105–114.
- Sayama, H. (2010). Swarm chemistry evolving. In *Artificial Life XII: Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pages 32–33.
- Sayama, H. (2012a). Morphologies of self-organizing swarms in 3d swarm chemistry. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, pages 577–584. ACM.
- Sayama, H. (2012b). Swarm-based morphogenetic artificial life. In Doursat, R., Sayama, H., and Michel, O., editors, *Morphogenetic Engineering, Understanding Complex Systems*, pages 191–208. Springer Berlin Heidelberg.
- Schmickl, T. and Hamann, H. (2011). Beeclust: A swarm algorithm derived from honeybees. *Bio-inspired Computing and Communication Networks*. CRC Press.
- Seeley, T., Visscher, P., and Passino, K. (2006). Group decision making in honey bee swarms: When 10,000 bees go house hunting, how do they cooperatively choose their new nesting site? *American Scientist*, pages 220–229.
- Thompson, D. (1917). *On growth and form*. Cambridge University Press.
- Wolfram, S. (1984). Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):1–35.

## An alife game to teach evolution of antibiotic resistance

Guillaume Beslon<sup>1</sup>, Bérénice Batut<sup>1,2</sup>, David P. Parsons<sup>1</sup>, Dominique Schneider<sup>3,4</sup>, and Carole Knibbe<sup>5</sup>

<sup>1</sup>Université de Lyon, CNRS, INRIA, INSA-Lyon, LIRIS, UMR5205, F-69621, France

<sup>2</sup>Université de Lyon, CNRS, LBBE, UMR5558, F-69622, France

<sup>3</sup>Laboratoire Adaptation et Pathogénie des Micro-organismes, Université Joseph Fourier Grenoble, F-38042, France

<sup>4</sup>CNRS UMR5163, F-38042 Grenoble cedex 9, France

<sup>5</sup>Université de Lyon, CNRS, INRIA, Université Lyon 1, LIRIS, UMR5205, F-69622, France  
guillaume.beslon@inria.fr

### Abstract

The emergence of antibiotic resistant bacteria is a major threat to public health and there is a constant need for education to limit dangerous practices. Here, we propose to use alife software to develop training media for the public and the physicians. On the basis of the Aevol model we have been developing for more than six years, we built a game in which players fight bacterial infections using antibiotics. In this game the bacteria can evolve resistance traits, making the infection more and more difficult to cure. The game has been tested with automatic treatment procedures, showing that it behaves correctly. It has been demonstrated during the French "Nuit des Chercheurs" in October 2012.

### Introduction

The rapid spread of antibiotic resistant bacteria is a growing threat to public health. Attempts to fight this threat include searching for new antibiotic molecules, understanding the evolutionary dynamics of resistance traits and organizing health care services to avoid dissemination of resistant bacteria. However, all specialists and authorities agree that the most important challenge to fight resistance is education. As Stuart B. Levy already argued in 2002: "Much work is needed on education of the consumer and the prescriber" Levy (2002). In many countries, large awareness campaigns were conducted, but even after ten years, (Bush, 2011) still claimed that "substantial increases in public education about bacteria and antibiotic importance are vitally important". Despite fundings from international and national agencies and the enrollment of non-profit organizations, the basic laws of antibiotic resistance are still very poorly known by the public, leading to maladapted usages that favor the spread of resistant strains.

One of the difficulties when teaching antibiotic resistance is that many factors are intertwined, leading to messages that might sound contradictory (e.g., limit antibiotic usage *but* systematically finish your treatment even though you feel you are cured!). To understand the resistance problem, one needs to grasp the entire evolutionary dynamics of antibiotic resistance, from the selection of resistant mutants to their

spread in a bacterial population (MacCallum, 2007). In particular, one must understand the paths that can lead to higher resistance levels (Almahmoud et al., 2009; Weinreich et al., 2006). These dynamics are far from trivial and are influenced by many parameters: the mutation rate, the population size, population bottlenecks or the fitness cost of resistance. As a matter of fact, the very principles of evolution are poorly known by the public and even by physicians.

To facilitate the understanding of the microbial world and dynamics, computer games have been developed like e-Bug (Lecky, 2011; Farrell, 2011), Bait (Kerr, 2005) or the virtual infection control simulation (Pulman and Shufflebottom, 2009). These programs emphasize the population dynamics of the resistant and susceptible bacterial strains with or without different antibiotics but they do not include evolution *per se* and cannot account for resistance appearance and increase. Although attempts to include evolutionary algorithms in games are numerous, they mainly focus on the evolution of avatar behavior (Grand and Cliff, 1997; Stanley et al., 2005) and often take many freedoms with the real evolutionary phenomena (Bohannon, 2008). We argue that artificial life evolutionary models have a huge potential to develop educational games and that too little work has been done in this direction, except for few occasional attempts (Adami Lab. and Beacon center, 2013; Miglino et al., 2012).

Here, we present the "Aevol Serious Game", based on the Aevol model of bacterial evolution. The game idea is quite simple: the player fights a series of bacterial infections with five different antibiotics: *prokarycin*, *microbicin*, *bactericin*, *bacillicin* and *aevolicin* (all antibiotic names but the last one have been chosen to enable the teacher to explain what is a "microbe"). The player controls which antibiotics are delivered and at what doses. The availability of five antibiotics allows to modulate the treatment strategies to fight the infections. Of course, at the beginning of the game, one can cure an infection very simply by setting one or more antibiotics to the maximum for a few generations. However, the artificial bacterial colony may evolve antibiotic resistance traits that make it more difficult to cure subsequent infections. Moreover, the acquired resistance traits directly depend on the



way antibiotics have been used previously and these effects cannot be cancelled. Thus, as the game goes on, it becomes increasingly difficult to kill the bacteria and to fight infectious diseases.

We first present how the evolving bacterial population is modeled. We then illustrate the model behavior with a simulated player who delivers antibiotics in various automatic ways. Finally, we report how non-scientific players used the game during “La Nuit des Chercheurs” in October 2012.

## Modeling the evolving bacterial population

The Aevol model (<http://www.aevol.fr>) was developed to study the evolution of genome structure and the influence of indirect selection pressures for robustness or evolvability, see Knibbe et al. (2007, 2008). We present here an adapted version of the model, aimed at teaching non-scientific public about antibiotic resistance in the context of evolution. The model is organized as a generational evolutionary algorithm, each generation consisting in three main steps: genome decoding, selection and reproduction with both local mutations and chromosomal rearrangements.

## From genotype to phenotype

The genotype-phenotype mapping in Aevol was inspired by the microbial transcription and translation processes. Each artificial organism owns a genome organized as a circular double-stranded binary string containing a variable number of genes separated by non-coding sequences (figure 1). A set of signaling sequences is used to identify the regions that will be transcribed into mRNAs and within those the ones that will be translated into proteins.

Transcription initiation and termination sites are directly inspired by bacterial genetics. In Aevol, we defined a promoter as a sequence close enough to a predefined consensus and a terminator as a short sequence able to form a stem-loop structure. When a promoter is found, the transcription proceeds until a terminator is reached, thus producing an mRNA whose expression level directly depends on the promoter quality.

Translation occurs when a ribosome-binding site is present on an mRNA, followed by a START codon. Then, the following sequence is read three bases (one codon) at a time, until an in-frame STOP codon is found. Each codon is translated into an abstract “amino-acid” using an artificial genetic code (figure 1).

An artificial chemistry (Dittrich et al., 2001) was defined to model the protein activity and the resulting phenotype. We defined an abstract, one-dimensional space  $\Omega = [0, 1]$  of possible cellular processes (therefore, in this model, a “cellular process” is a real number). Each protein can either realize or inhibit a particular set of these biological processes with a certain efficacy. For simplicity, we use piecewise-linear functions with a symmetric, triangular shape. Hence

the activity of a protein can be fully characterized by the position  $m$  of the triangle on the axis, its half-width  $w$  and its height  $h$ .

A protein’s primary sequence is viewed as three interlaced binary sequences that code for  $m$ ,  $w$  and  $h$  values (see figure 1). Small mutations in the coding sequence will change these parameters, resulting in a modification of the protein activity. Once all proteins encoded by a given genotype have been identified and characterized, their activities are combined into a global fuzzy set representing the individual’s phenotype  $P$ . The possibility distribution of  $P$ , called  $f_P$ , indicates to what extent the individual can realize each abstract cellular process.

For the game, the interval  $\Omega = [0, 1]$  of cellular processes is divided into 7 subintervals (figures 1 and 3). The first ( $\Omega_0$ ) and last ones ( $\Omega_6$ ) represent sets of abstract metabolic processes and they are used to determine the performance of the individual in the competition for reproduction (see below). The five intermediate zones ( $\Omega_1$  to  $\Omega_5$ ) correspond to resistance traits respectively against the five antibiotics. The proximity of these traits along the functional axis facilitates multi-drug resistance (a common feature in bacteria) through pleiotropy since a single triangle can span 2 contiguous zones.

## Environment, fitness measure, death and competition

In Aevol, the environment is represented by a phenotypic target: a fuzzy set  $E$  defined on  $\Omega$  whose possibility distribution  $f_E$  indicates the optimal degree of possibility for each “metabolic process”. To evaluate an individual, we compare its phenotype  $P$  to the optimal one  $E$ . The “gap with target”  $g$  is computed as the geometric area between these two sets on the “metabolic” subintervals (figure 1):  $g = \int_{\Omega_0} |f_E(x) - f_P(x)| dx + \int_{\Omega_6} |f_E(x) - f_P(x)| dx$ . The lower the gap, the fitter the individual. This penalizes both the under- and over-realization of each metabolic process.

Cells can die with a probability  $P_{death} = 20 \times g$  if  $g < 0.05$ , or  $P_{death} = 1$  if  $g \geq 0.05$ , implying that some mutations are lethal. This mortality process, not present in the original Aevol model, was introduced in the game because antibiotics may cause severe population bottlenecks, which are known to make selection less efficient and allow mildly deleterious mutations to accumulate in the population (an effect used by microbiologists in “mutation accumulation” experiments (Korona, 2004)). Making highly deleterious mutations in the game actually lethal prevents them from propagating during the population bottlenecks. In the runs below, typically around 20% of the individuals die at each generation through this mortality process.

Cells can also die because of an antibiotic treatment. Although antibiotics can act on different bacterial subsystems (Normark and Normark, 2002), in this first version of the game, we simply simulated an effect on cell mortality de-

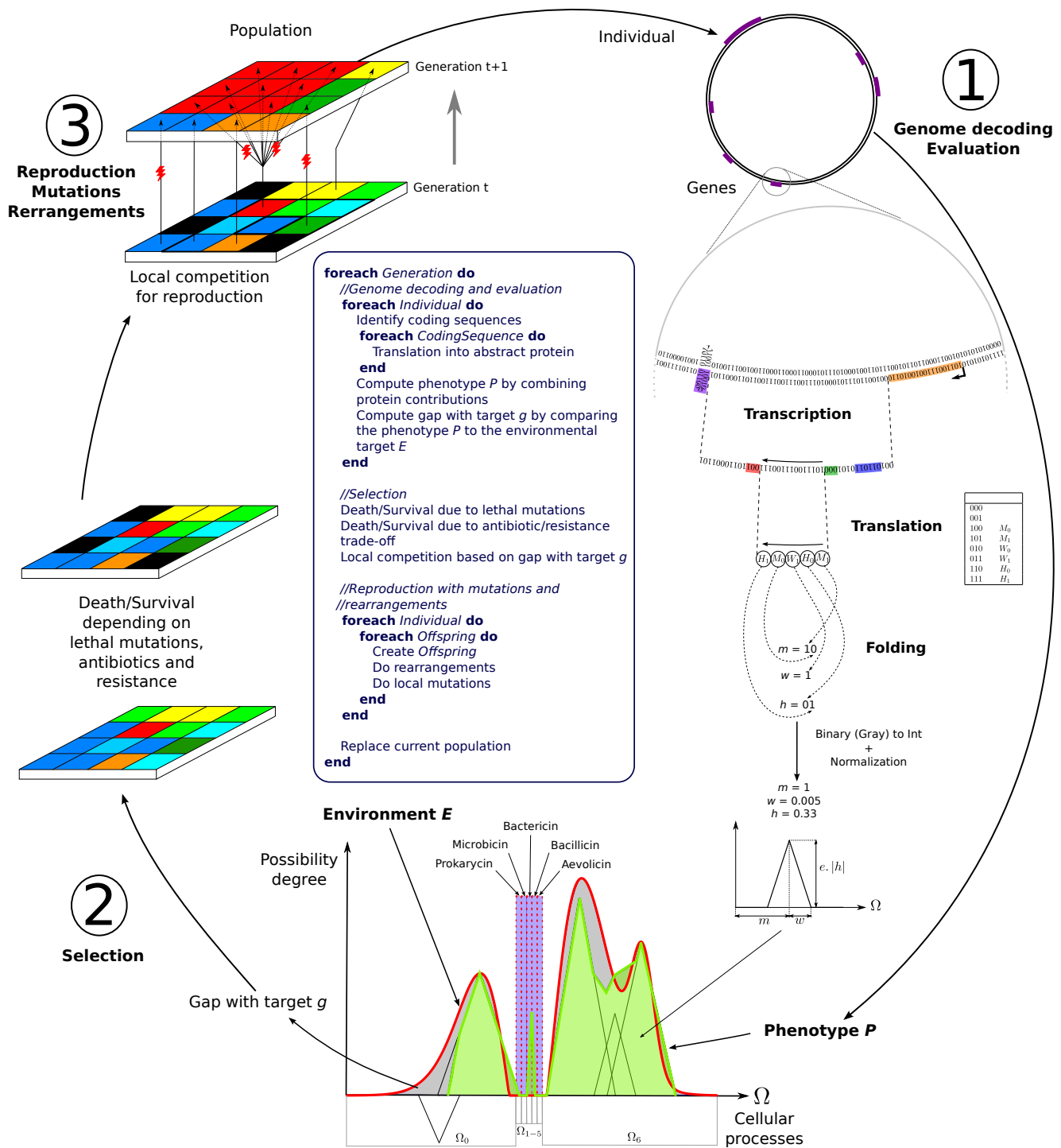


Figure 1: Graphical representation of the AevoI algorithm with antibiotics. The algorithm iterates three main steps: (1) genome decoding and evaluation, (2) selection of the best individuals and (3) reproduction with mutations and rearrangements. See the main text for details. The lightnings correspond to mutations and rearrangements undergone during reproduction. Cells on the grid are colored according to gap with target, red cells being those with lowest  $g$  and blue cells being the higher ones. The dead cells are the black cells. The violet zone on the cellular process axis ( $\Omega_1$  to  $\Omega_5$ ) correspond to resistance traits (one zone per antibiotic).

pending on antibiotic dosage. When a cell receives an antibiotic  $i$  with a dosage  $\alpha_i$  ( $0 \leq \alpha_i \leq 1$ ), it has a probability  $P_{antibio.i} = 0.9 \times \alpha_i$  to die<sup>1</sup> except if it is protected by a resistance trait. The resistance traits to the antibiotics are deduced from the phenotypic function within the  $\Omega_1$  to  $\Omega_5$  zones (one zone per antibiotic, figure 1). The probability to survive a treatment with a given antibiotic  $i$  is:  $P_{resist.i} = 1 - 100 \int_{\Omega_i} |0.5 - f_P(x)| dx$ . There is no direct cost to resistance. However, since resistance is generally acquired through mutation of existing genes, indirect costs are very likely to occur. If more than one antibiotics are delivered simultaneously, the same process is applied for all the antibiotics.

Cells are placed on a  $40 \times 40$  grid (Misevic et al., 2012). Each grid spot contains either a living or dead cell (figures 1 and 3). Local competition for reproduction takes place between living cells only. The population is entirely renewed at each generation. Specifically, each grid spot at generation  $t + 1$  is filled with an offspring from one of the neighboring living cells at generation  $t$  (note that this offspring can die immediately, e.g. if it undergoes a lethal mutation). The cell that will produce the offspring is drawn according to a reproduction probability  $P_{reprod}$  that decreases with  $g$ . If none of the 9 neighboring spots contained a living cell at generation  $t$ , the spot is left unchanged.

### Reproduction, local mutations, rearrangements

During their replication, genomes can undergo both local mutations (single nucleotide substitutions, and insertions or deletions of 1 to 6 bp) and chromosomal rearrangements (duplications, deletions, translocations and inversions). Here, all local mutation occurred with probability  $\mu_{mut} = 1 \times 10^{-5}$  per bp per generation and all rearrangements with probability  $\mu_{rear} = 1 \times 10^{-6}$  per bp per generation. Not all mutational events have a phenotypic effect. For example, a mutation in a region that is not transcribed will most probably be neutral (except if it occurs in a promoter). Because Aevol allows for gene duplication and divergence, it can evolve new functions (e.g. antibiotic resistance traits) and not only modify existing ones.

### Evolution of wild-type strains

Before the game itself, the software was used to evolve a “wild-type strain” in the same environment as used for the game without antibiotics. Here, we used a strain that evolved for 100,000 generations. Its genome is 80,920 base-pair long and has 106 coding sequences present on 101 coding mRNAs. It is well adapted to its environment ( $g \approx 0.006$ ), ensuring that slightly deleterious mutations can accumulate without being immediately lethal.

<sup>1</sup>The 0.9 factor prevents the population from being killed all at once: at maximum dosage the infection is cured in 8 to 10 generations if no resistance evolves. Note that, for pedagogical reasons,  $\alpha$  is always displayed as percentages of the maximal dose.

### Simulating infection

During the first steps of the game, driving the population to extinction is easy since the wild-type is highly susceptible to antibiotics. In such case, the game can go on with a reinfection (triggered by the player). Then the cell with the lowest  $g$  switches from dead to alive. The resistance traits are thus more difficult to maintain since they are not considered in the choice of the “resuscitated” cell, thereby mimicking an infection from an antibiotic-free environment. The resistance traits are actually eliminated unless they were carried by the individual with the best  $g$ , a situation that usually occurs when the resistance traits were useful for a very long period (long enough to enable fixation of these traits despite their mutational load). After reinfection, the germ progressively colonizes all the grid, showing circular patterns where central cells accumulate fewer mutations than peripheral ones (figure 2). This pattern reproduces a known property of population expansion: mutations accumulate on the expanding fronts (Excoffier and Ray, 2008).

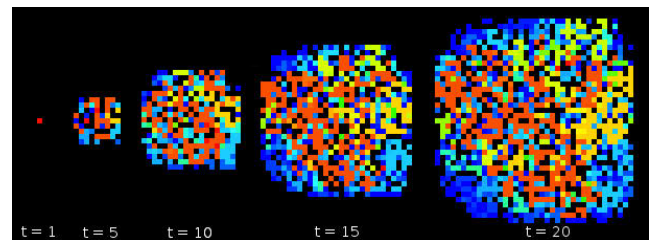


Figure 2: Population expansion during an infection process (here at  $t = 1, 5, 10, 15$  and  $20$  generations after the reinfection). Cells are colored according to fitness, red cells being those with highest fitness (lowest  $g$ ) and blue cells being the worst ones. The circular pattern is clearest at  $t = 15$  and  $t = 20$  where many blue cells are observed in the periphery of the infection, while the central zone contains mainly orange and red ones.

### Graphical outputs

The Aevol game is not designed to be used by the player alone but with a supervisor who can explain in real time the effects of the treatment and advise the player. The graphical outputs (figure 3) were designed accordingly. On the left panel, the player can see the whole population on the grid, each living cell being represented by a square, colored according to the gap to target  $g$  of the individual. This panel also displays the current antibiotic dosages that can be changed through the keyboard. It allows the player to directly perceive the effect of the antibiotic on the number of living cells. The right panel displays three different views of the current best living cell. Two of them represent its genome, with either the transcribed (top left) or translated sequences (top right). The third one (bottom) shows its phenotype (in green) including the resistance traits (for clarity

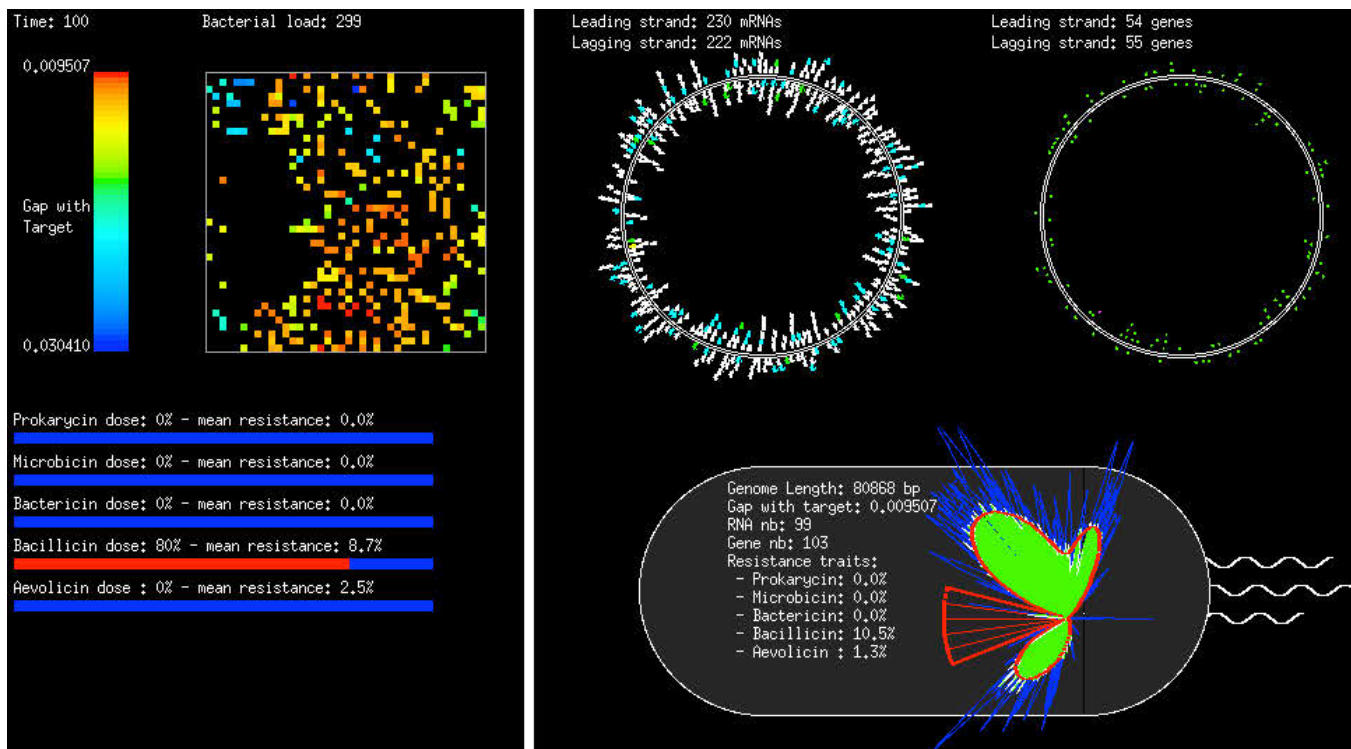


Figure 3: Screen capture of the Aevol game. The left panel presents bacteria population with a color code indicating their fitness (red cells being the fittest and blue the least fit). It also presents the current concentration of the five antibiotics (here, *bacillicin* is given at a 80% dose). The right panel presents the fittest individual's transcriptome (top left), genome (top right) and phenotype (bottom). The phenotype is represented in polar coordinates (green surface) together with the target function (red curve) and the five resistance sub-functions (red triangles). Here, the fittest individual became resistant to *bacillicin* after less than 100 generations (but resistance is not high enough to be fixed yet). Finally, the phenotypes of all living cells are represented with blue lines, allowing an estimation of the population diversity.

and esthetic reasons,  $\Omega$  is displayed in polar coordinates). It also shows the environmental target (red curve), the optimal shapes for the five resistance traits (red triangles), and the phenotypes of all living cells (blue lines). This allows the player to estimate the resistance traits in both the best individual and the rest of the population and thus to detect clonal interference between the metabolic and the resistance genes.

### Behavior with an automatic player

Although the Aevol game was not developed to *study* but to *teach* the emergence of resistant traits, we show here the behavior of the model with a simulated player that delivers antibiotics in various automatic ways. This study enables to verify that the behavior of the model is realistic enough to be used to educate people. It is also important for teachers that can use it to prepare himself to comment the players' actions.

### Effects of antibiotic dose

For each antibiotic, the wild-type strain was given a constant antibiotic delivery  $\alpha$ , varying between 50% and 100%.

Each time the antibiotic treatment resulted in eradicating the bacterial population, a new infection was automatically triggered and the antibiotic treatment was momentarily stopped until the population grew over 1000 living cells (the carrying capacity of the environment being 1600 individuals). For each antibiotic and dosage, we conducted the simulation until the resistance trait was considered to be fixed in the population (i.e. the resistance of the best living cell was sufficient to make it resistant to a maximum antibiotic dose). We measured both the number of successes of the antibiotic (number of infections driven to extinction) and the number of elapsed generations before the resistance criterion was met.

Figure 4 shows that antibiotics cannot drive the infections to extinction for doses below 80%. The effect of antibiotic dosage is intuitively obvious: the higher the dosage, the higher the probability to eradicate the infection.

Figure 5 shows whether the resistance criterion is met and if so, how fast, depending on the dose. Three ranges of dosages can be distinguished by comparing figures 4 and 5. For  $\alpha < 60\%$  (low dosage), the treatment fails and no resistance is acquired, most likely because the fitness cost



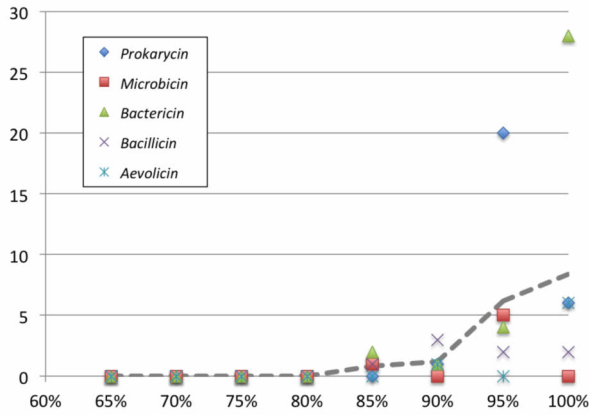


Figure 4: Number of infections successfully cured, as a function of the antibiotics dosage  $\alpha$  (dashed curve: mean).

associated with the resistance mutation is too high compared to the benefit it confers. For  $60\% \geq \alpha \geq 80\%$  (intermediate dosage), the treatment also fails but resistance can be acquired, which confirms the risk associated with sublethal dosages. For  $\alpha \geq 85\%$  (high dosage), although the treatment succeeds several times, all populations eventually acquired resistance. Increasing dosage generally speeds up resistance acquisition, but the precise relationship is antibiotic-dependent. More experimental replicates with different random seeds and wild-types would be needed to test whether this effect is random, genome-dependent (i.e. dependent on the probability to find a favorable mutation in the genome) or antibiotic-dependent (i.e. dependent on the position of the resistance trait in  $\Omega$ ).

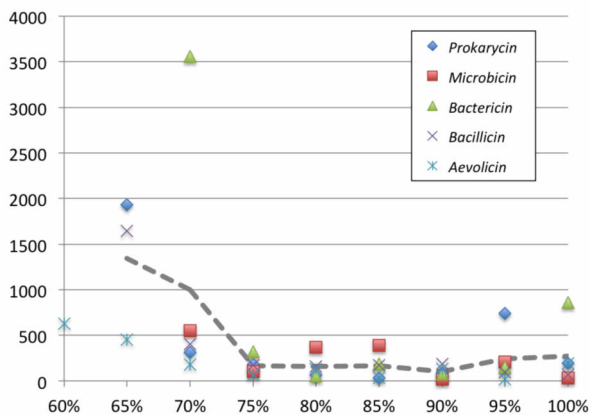


Figure 5: Number of generations elapsed before the resistance criterion was met, as a function of the antibiotics dosage  $\alpha$  (dashed curve: mean). For  $\alpha = 60\%$ , only *aevollicin* led to resistance. For  $\alpha = 65\%$ , 3 of the 5 antibiotics led to a resistance.

These first results validate the game as a realistic tool to educate people for a correct use of antibiotics. They show

that, in the model, bacteria can acquire resistance traits and that the time to fixation of these traits depends on the way antibiotics are used. Moreover, the behavior of the five antibiotics is globally coherent, with slight differences that complexifies the game.

### Effect of treatment timing

For each of the five antibiotics, the wild-type strain has been submitted to intermittent treatments (with a 100% dosage). The dose delivery randomly switches from treatment to non-treatment using a Poisson process with a switching probability ranging from  $1/2$  to  $1/11$  at each generation. In all cases, the mean antibiotic delivery over a long period of time is 50%. Figure 6 shows that the treatment timing has no effect on resistance acquisition. Similarly, there is no effect on the probability of success of the treatment (data not shown).

In contrast to the previous results, here the bacteria quickly acquire resistance in all cases (the maximum delay being 1318 generations) while, for the same mean dosage (50%), a constant treatment never leads to resistance acquisition. Besides, intermittent treatment leads to a mean of 4.3 defeated infections before the resistance is fixed, regardless of the frequency of switches<sup>2</sup>, while such a low dose could not eradicate the infection with a constant treatment. This is probably an indirect effect of the fluctuation of population size when the infection is treated in an intermittent way. Indeed, a constant treatment with low doses leads to a small but constant population size (around 650 individuals for a 50% dose). In contrast, an intermittent treatment leads to huge variations in the population size (see figure 7 for an example).

Population genetics theory states that in case of variations in the population size, the effective population size is given by the harmonic mean over time of the real population size. Here, for a mean switching probability of  $1/3$  and before any resistance trait acquisition, the population size varies between 1 and 1250 individuals (mean value around 350 individuals) but the harmonic mean of the population size (excluding the periods of infection when no antibiotic is delivered) is around 18 individuals! The high level of genetic drift can hence explain that resistant individuals (that generally also carry deleterious mutations) sometimes manage to reproduce, thereby favouring the spread of resistance. Moreover, when the treatment temporarily stops, the population can expand. As figure 2 shows, this leads to mutation accumulation in the population. Thus, intermittent treatments also increase the apparent mutation rate (although the spontaneous mutation rate remains constant), thus increasing the population diversity and hence favouring the emergence of

<sup>2</sup>The different antibiotics do, however, behave differently: *prokarycin* leads to 4 successes on average, *microbicin* 5.9; *bactericin* 7.6; *bacillicin* 3.3 and *aevollicin* 0.8. This is consistent with the previous section, *aevollicin*, in particular, being more prone to resistance traits acquisition.

resistant mutants.

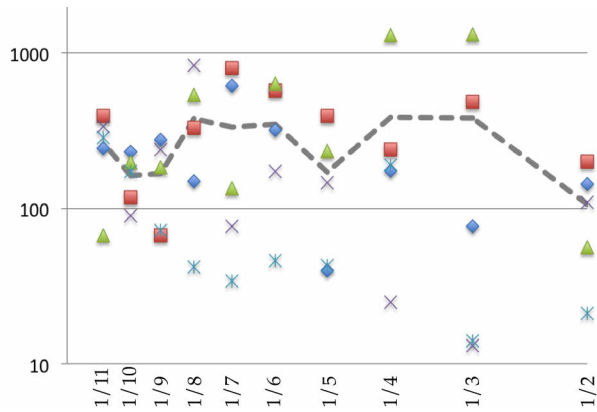


Figure 6: Number of generations elapsed before the resistance criterion was met, as a function of the antibiotics treatment dynamics. The antibiotic treatment switches between on and off at each generation with a probability ranging from  $1/2$  to  $1/11$ . See figure 4 or 5 for the legend.

These differences between intermittent high dose treatment and constant small dose treatment (for similar mean) shows the importance of taking into account evolutionary effects like population bottlenecks and spatial expansion when dealing with antibiotic resistance.

### Effect of pharmacodynamics and adherence

In the first two experiments, the antibiotic dosage was either constant or piecewise-constant over time. However, in a real situation, the antibiotic delivery depends on many physiological parameters such as means of delivery and half-life of the drug. The Aevol game can be used to teach the effects of these parameters and to introduce a patient's adherence to the prescribed treatment in the model, that is, the fraction of scheduled doses taken. Indeed, it has been shown that adherence can have a strong influence on resistance emergence, at least during antiviral therapy (Rosenbloom et al., 2012). Here, we show the behavior of an antibiotic treatment with *bacillicin*. The drug was given to the patient at 40% doses with one dose every 9 generations (the maximum drug concentration still being 100%). Five percent of the drug was degraded at each generation (drug half-life: 21.5 generations).

Figure 7, top panel, shows how drug concentration evolves during a treatment with perfect adherence, and the resulting effect on the bacterial load. Since the evolutionary process is included in the simulation, we can study the influence of drug dynamics on the emergence of a resistant mutant in relation to the population expansion that occurs regularly during the antibiotic delivery. The game can also be used to show the impact of patient drug-taking behavior on the dynamics of both the bacterial population and emergence of resistant strains. Figure 7, bottom panel, shows

the drug concentration, bacterial population and resistance to *bacillicin* when the patient randomly misses one dose out of four. Here, after only two missed doses (doses 4 and 7), the infection duration (initially 68 generations) is substantially longer. This also increases the risk for resistance to emerge before the end of the treatment, which in the present case does happen. Note that the simulation presented here was specifically chosen to illustrate the emergence of resistance due to a bad adherence. However, systematic experiments have not been performed yet on this question and no conclusion can be taken at this stage.

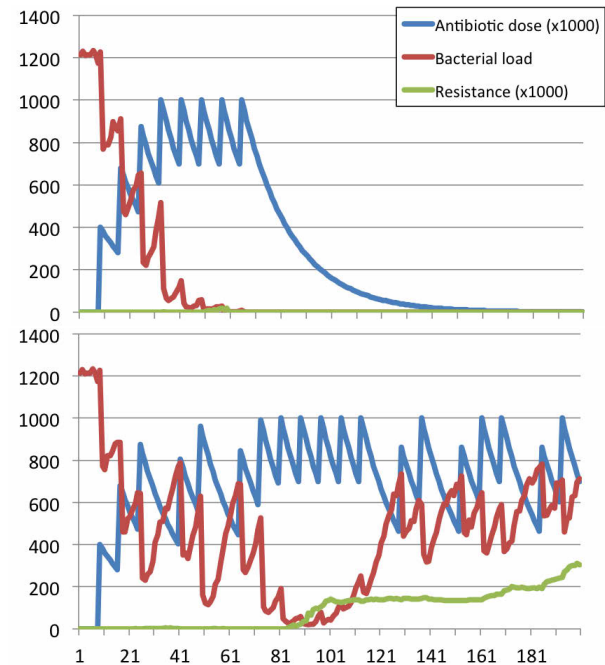


Figure 7: Effect of pharmacodynamic (top panel) and adherence (bottom panel) on bacterial load and antibiotics resistance (see main text for details).

### Conclusion and future work

The Aevol game was used in real conditions during “La Nuit des Chercheurs 2012” (October 2012), one of the main public science events in France. The simulation ran for three hours using a beam projector, and around 40 people from 10 to 65 years old played with it (the game was also successfully tested with younger children but in a smaller audience). After playing with the game, visitors wandered around the place, discussed with other researchers, and came back later to see how the game evolved and possibly play again. The result was a clear success: during the whole event, the bacterial population progressively acquired resistance to the five antibiotics. By the end of the experiment, complex treatments had become mandatory to fight the infections. Visitors could witness that the misuse of antibiotics during their first attempts had strongly influenced the evolution of



the population in the long term and created the subsequent difficulty for the next players. This shows the importance of a global tracing of the game, allowing the players to visualize the effects of previous treatments and detect *a posteriori* those having succeeded and those resulting in resistance acquisition. This experiment also showed that the game cannot be used without a supervisor to explain the behavior of the population and provide a minimal basis of evolution and genetics.

Although preliminary, the Aevol game provides a proof of concept that life models can be used to teach people difficult but relevant scientific questions. It also opens the door to many new developments. First, although a game is not a model, it provides the opportunity to compare the simulated dynamics with what is observed in *in vivo* experimental evolution (Hindré et al., 2012). This will require a more precise model of antibiotic action on the cell and of antibiotic resistance traits. Second, there are ample opportunities for improvement of the game. An important direction would be to enable the player to browse the whole population (not only the best individual) and to visualize *a posteriori* the entire course of evolution, starting from the beginning of the antibiotic treatment. Antibiotic delivery can also be improved. Here the antibiotic dose is directly fixed by the player (figure 3). One may ask the player for a prescription (i.e. dose and scheduling of antibiotic uptakes) and include pharmacodynamics in the model. Moreover, in the current version, antibiotics are harmless for the patient and can be used at maximum dosage without deleterious effects. Adding such effects (which are well documented for this kind of drugs) will add complexity to the game. Finally, an island model can be introduced in the game. Different islands could represent different patients as well as the global environment, thus enabling complex infection dynamics and multi-player gaming (e.g. on the Internet).

### Acknowledgements

We would like to thank Dusan Misevic and Antoine Frenoy for their collaboration on Aevol development. This research was funded by the Centre National de la Recherche Scientifique (interdisciplinary programs PEPS and PEP2), the Institut Rhône-Alpin des Systèmes Complexes (IXXI) and the FINOVI foundation.

### References

- Adami Lab. and Beacon center (2013). The evolve & conquer project. <http://adamilab.msu.edu/evolve-and-conquer/>.
- Almahmoud, I., Kay, E., Schneider, D., and Maurin, M. (2009). Mutational paths towards increased fluoroquinolone resistance in *Legionella pneumophila*. *J. Antimicrob. Chemother.*, 64(2):284–293.
- Bohannon, J. (2008). Flunking spore. *Science*, 322(5901):531.
- Bush, K. et al. (2011). Tackling antibiotic resistance. *Nat Rev Microbiol*, 9(12):894–896.
- Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries-a review. *Artif Life*, 7(3):225–275.
- Excoffier, L. and Ray, N. (2008). Surfing during population expansions promotes genetic revolutions and structuration. *Trends Ecol Evol*, 23(7):347–351.
- Farrell, D. et al. (2011). Developing e-bug web games to teach microbiology. *J Antimicrob Chemother*, 66 Suppl 5.
- Grand, S. and Cliff, D. (1997). Creatures: Entertainment Software Agents with Artificial Life. *Autonomous Agents and Multi-Agent Systems*, 1(1):39–57.
- Hindré, T., Knibbe, C., Beslon, G., and Schneider, D. (2012). New insights into bacterial adaptation through *in vivo* and *in silico* experimental evolution. *Nat. Rev. Microb.*, 10:352–365.
- Kerr, G. (2005). Bait: Bacteria - antibiotic interaction tool. *ERCIM News*, 60:2.
- Knibbe, C., Coulon, A., Mazet, O., Fayard, J.-M., and Beslon, G. (2007). A long-term evolutionary pressure on the amount of noncoding DNA. *Mol. Biol. Evol.*, 24(10):2344–2353.
- Knibbe, C., Fayard, J.-M., and Beslon, G. (2008). The topology of the protein network influences the dynamics of gene order: from systems biology to a systemic understanding of evolution. *Artif. Life*, 14(1):149–156.
- Korona, R. (2004). Experimental studies of deleterious mutation in *Saccharomyces cerevisiae*. *Res Microbiol*, 155:301–310.
- Lecky, D. et al. (2011). Development of an educational resource on microbes, hygiene and prudent antibiotic use for junior and senior school children. *J Antimicrob Chemother*, 66 Suppl 5.
- Levy, S. (2002). Factors impacting on the problem of antibiotic resistance. *J. Antimicrob. Chemother.*, 49:25–30.
- MacCallum, C. (2007). Does medicine without evolution make sense? *PLoS Biology*, 5(4):e112.
- Miglino, O., Gigliotta, O., Schembri, M., and Di Ferdinando, A. (2012). Collective adaptive agents as techniques to build-up edutainment systems. In *Italian Workshop on Artificial Life and Evolutionary Computation*.
- Misevic, D., Frenoy, A., Parsons, D. P., and Taddei, F. (2012). Effects of public good properties on the evolution of cooperation. In *Proceedings of Artificial Life XIII*, pages 218–225.
- Normark, B. H. and Normark, S. (2002). Evolution and spread of antibiotic resistance. *J. Intern. Med.*, 252:91–106.
- Pulman, A. and Shufflebottom, M. (2009). A virtual infection control simulation—the development of a serious game in the health care sector. *Networks*, 97:97–106.
- Rosenbloom, D., Hill, A., Rabi, S., and Siliciano, R.F. and Nowak, M. (2012). Antiretroviral dynamics determines hiv evolution and predicts therapy outcome. *Nature Medicine*, 18(9):1378–1385.
- Stanley, K., Bryant, B., and Miikkulainen, R. (2005). Real-time neuroevolution in the nero video game. *Evolutionary Computation, IEEE Transactions on*, 9(6):653–668.
- Weinreich, D., Delaney, N., Depristo, M., and Hartl, D. (2006). Darwinian evolution can follow only very few mutational paths to fitter proteins. *Science*, 312:111–114.

# Analysis of Ultrastability in Small Dynamical Recurrent Neural Networks

Eduardo J. Izquierdo<sup>1</sup>, Miguel Aguilera<sup>2</sup>, Randall D. Beer<sup>1</sup>

<sup>1</sup> Indiana University Bloomington, IN, U.S.

<sup>2</sup> Universidad de Zaragoza, Spain  
edizque@indiana.edu

## Abstract

This paper reconsiders Ashby’s framework of adaptation within the context of dynamical neural networks. Agents are evolved to behave as an ultrastable dynamical system, without imposing a priori the nature of the behavior-changing mechanisms, or the strategy to explore the space of possible dynamics in the system. We analyze the resulting networks using dynamical systems theory for some of the simplest conditions. The picture that emerges from our analysis generalizes the idea of ultrastable mechanisms.

## Introduction

Organisms change their behavior in an attempt to remain adapted to their interaction with the environment, in order to maintain their self-constitution. W. Ross Ashby developed a theoretical framework to understand operationally this tendency towards ultrastability in organisms (Ashby, 1960). His approach considered the role of what can be interpreted as proxies to the essential variables of the organism - variables that correlate with its physiological well-being and that are directly available to the organism (e.g., body temperature, sugar level, oxygen intake). A behavior is ultrasatable if it conserves the proxies to the essential variables of the organism within some physiological limits under changing conditions. This paper explores the nature of the dynamics of ultrastable systems in artificially evolved neural networks.

Ashby’s framework of ultrastability is still central to much of the work on adaptive behavior, but many of the details of the implementation in neural networks have not been studied in enough depth (Harvey, 2008). The mechanism of adaptation Ashby studied was hand-designed specifically to find stable agent-environment interactions during perturbations. There are two aspects of his original framework that we seek to re-examine. First, the process of adaptation depends on a strict separation between the parameter-changing mechanisms and the behavior-producing mechanisms. In doing so, it relies heavily on the stochastic, switch-like nature of the behavior-changing mechanisms. Second, the framework of adaptation treats the projection of the critical boundary on the space of internal dynamics of the agent as a function only of the constitution of the agent.

In this paper we ask: what are the agent-environment conditions that allow artificial evolution to find neural networks with ultrastable properties? And, what are some of the ways in which ultrastability specified at the behavioral level can be implemented dynamically at the neural network level? We are interested in how a dynamical system can explore the richness of its state space, navigating through the different regions of its phase-portrait. The goal of this paper is to analyze neural networks evolved to be ultrastable in an attempt to extend and generalize Ashby’s framework of adaptation.

## Ashby’s Ultrastable Mechanism

Ashby based his framework around the basic assumption that organisms change their behavior by learning, so that the later behavior is better adapted to their environment than the earlier. He distinguished between two kinds of nervous system activity: hardwired reflex and learned behavior. In his framework (Figure 1A), the agent ( $\mathcal{A}$ ) is defined by two subsystems:  $\mathcal{R}$ , responsible for the ongoing reflexive behavior; and  $\mathcal{S}$ , responsible for regulating the behavior of  $\mathcal{R}$ . There are also a set of variables,  $\mathcal{P}$ , that serve as proxies to the organism’s essential variables ( $\mathcal{V}$ ). The environment ( $\mathcal{E}$ ) is defined as a system whose variables affect the organism through coupling and which are in turn affected by it.

For the organism to survive, its essential variables ( $\mathcal{V}$ ) must be kept within viable limits (gray area, Figure 1B). The ‘constitutive viable region’ maps to an ‘internal viable region,’  $\mathcal{M}(\mathcal{V})$  (gray areas, Figure 1C). When the agent’s internal dynamics are inside this region, the essential variables are kept within viable limits. When the internal dynamics are outside this region, the agent-environment dynamics have an effect on the proxies to the essential variables of the system. When the proxies are outside the appropriate ranges for the organism, they introduce parametric changes in  $\mathcal{S}$ . The configuration of  $\mathcal{S}$  influences the agent-environment interaction through  $\mathcal{R}$ . Ashby proposed a step-function as a mechanism for  $\mathcal{S}$ . In Ashby’s view, adaptation occurs as the system randomly flips through the catalog of  $\mathcal{R}$  dynamics, until it stumbles across one that is stable for the current environment.

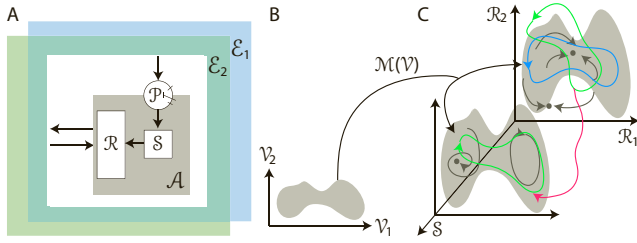


Figure 1: Ultrastable system (adapted from Ashby, 1960). [A] Architecture: Environments ( $\mathcal{E}_i$ ); agent ( $\mathcal{A}$ ); reactive system ( $\mathcal{R}$ ); system in charge of learning ( $\mathcal{S}$ ); and proxy for the essential variables ( $\mathcal{P}$ ). [B] Region of viability (gray area) over the agent's essential variables ( $\mathcal{V}_i$ ), defined by the agent's constitution. [C] Agent's state space ( $\mathcal{R}$  and  $\mathcal{S}$ ). Gray attractors and limit cycles depict autonomous dynamics of  $\mathcal{R}$ . Blue and green trajectories depict dynamics of the system when coupled to environments  $\mathcal{E}_1$  and  $\mathcal{E}_2$ , respectively. Gray areas depict the internal viable regions, according to  $\mathcal{M}(\mathcal{V})$ . The magenta trajectory represents the state of the system during adaptation.

### Behaviorally Ultrastable Dynamical System

This paper extends the framework of ultrastability by reconsidering it from a behavioral level. There are two aspects of Ashby's framework that we re-examine.

First, in Ashby's mechanism and the more recent simulation models that have stemmed from his work (Di Paolo, 2000; Iizuka and Di Paolo, 2008; Herrmann et al., 2004), there is a strict division between the parts of the system that continuously interact with the environment, and the parts that change the internal organization of the system itself. As with previous work on modeling learning without synaptic plasticity (Yamauchi and Beer, 1994; Izquierdo et al., 2008), we consider a broader definition of an ultrastable system by removing the *a priori* distinction between the reactive and the learning components within the agent (Figure 2A). This allows us to explore some of the ways in which ultrastability specified at the behavioral level can be implemented at the neural network level.

Second, in Ashby's framework, the internal viable region (gray areas, Figure 1C) is a function  $\mathcal{M}(\mathcal{V})$  solely of the constitutive viable region (gray area, Figure 1B). When coupled, the agent-environment produce a dynamic which is specific to that environment (blue trajectory, Figure 1C). A change of environment modifies the coupled dynamics (green trajectory in the back, Figure 1C), but not the internal viable region. Therefore, a change in the agent-environment interaction can drive the internal dynamics of the agent outside its internal viable region, triggering changes in  $\mathcal{S}$  (magenta trajectory, Figure 1C), until a new coupled dynamic is found (green trajectory in the front, Figure 1C). This perspective is problematic because it assumes the internal dynamics of the agent must be within the same internal vi-

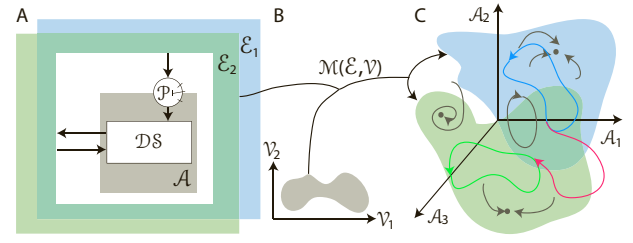


Figure 2: Extended ultrastable system. [A] Architecture: Environments ( $\mathcal{E}_i$ ); agent ( $\mathcal{A}$ ); internal dynamics ( $\mathcal{DS}$ ); and proxy for the essential variables ( $\mathcal{P}$ ). [B] Region of viability (gray area) over the agent's essential variables ( $\mathcal{V}_i$ ). [C] Agent's state space ( $\mathcal{A}$ ). Gray attractors and limit cycles depict the autonomous dynamics of  $\mathcal{A}$ . Blue and green trajectories depict the dynamics of the system when coupled to environments  $\mathcal{E}_1$  and  $\mathcal{E}_2$ , respectively. Colored volumes represent the internal viable regions,  $\mathcal{M}(\mathcal{E}, \mathcal{V})$ , colored according to the environment. Magenta trajectory represents the state of the system during adaptation.

able region for the two different behaviors. We consider a broader condition (Figure 2), where the internal viable region is a function of both the constitutive viable region and the environment,  $\mathcal{M}(\mathcal{E}, \mathcal{V})$ . Like in Ashby's case, we assume the constitutive viable region is defined over the essential variables by the organization of the organism (Figure 2B). However, when the organism is situated, the internal dynamics required to maintain the essential variables within their viability constraint changes as a function of the environment (blue and green regions, Figure 2C).

The picture of ultrastability in a dynamical system that emerges is different from Ashby's and subsequent models (Figure 2C). Assume a certain dynamics of the autonomous agent is suitable to maintain its essential variables within their viability constraints for a certain environment,  $\mathcal{E}_1$ , such that the coupled agent-environment interaction (blue trajectory) remains within the internal viable region (blue volume). Assume also a change to the environment,  $\mathcal{E}_2$ , such that the agent dynamics required to survive is different from the previous one (green volume). If the new coupled dynamics fall outside the new internal viable region, the agent's viability is threatened, setting off the proxies to the essential variables. The proxies drive the organization of the system towards an exploratory regime (magenta trace), until the behavior regains viability. The system regains viability by finding an internal dynamic that when coupled to the environment remains within the internal viable region.

In this view, a behaviorally ultrastable system must have three properties: (1) a rich reservoir of autonomous dynamics; (2) an exploratory regime capable of navigating the repertoire of autonomous dynamics; and (3) the ability to integrate those two regimes by modulating some aspect of the system via the proxies to the essential variables.

## Methods

We propose to study ultrastability by studying a simplified version of the extended view described above. We take for granted the self-constitution of the agent and the existence of a viability region. We also abstract away the agent-environment interaction. We assume the existence of  $N$  different environments such that, when combined with the agent's viability, they produce a series of internal viable regions. The organism is modeled as a network of  $N$  interacting neurons with  $C$  constrained ones and one proxy to the essential variables that can affect the whole network. The constrained neurons, in interaction with the environment, determine the state of the proxy to the essential variable. We artificially evolve these simple neural networks to maintain their proxies to the essential variables within viable ranges.

**Neural network** We used continuous-time recurrent neural networks (CTRNNs) as a model of the organism's internal dynamics. Each component in the network is governed by the following state equation (Beer, 1995):

$$\tau_i \dot{y}_i = -y_i + \sum_{j=1}^N w_{ji} \sigma(g_i(y_j + \theta_j)) + I_i \quad (1)$$

where  $y$  is the activation of each node,  $\tau$  is its time constant,  $w_{ji}$  is the strength of the connection from the  $j$ th to the  $i$ th node,  $g$  is a gain term,  $\theta$  is a bias term,  $\sigma(x) = 1/(1 + e^x)$  is the standard logistic activation function,  $I_i$  represents the external input to the neuron, and  $N$  represents the number of nodes in the network. The model does not include any form of synaptic plasticity mechanisms.

**Agent-environment interaction** For an adaptive organism, different environments are likely to require different internal dynamics. We abstract the agent-environment interaction by imposing arbitrary 'internal viable regions' over the neuron output space. Each of those regions is considered adaptive to a particular environment. That is, the dynamics available within this region of the phase-portrait are capable of maintaining the essential variables of the organism within their viability constraints. For simplicity, each environment is associated with one of the corners of the output space of the constrained neurons. The agent is considered adapted to the environment if its state in neural output is a distance less than 0.49 away from its corner.

**Proxy to the essential variable** According to the idealized rules above, at any point in time the agent can be adapted ( $a = 0$ ) or unadapted ( $a = 1$ ) to the environment. When the agent is adapted, the proxy to the essential variable rests at 0.0. When the agent is unadapted, the proxy to the essential variable increases over time, until it reaches one, according to the following state equation:  $\tau \dot{E} = -E + a$ . The proxy to the essential variable can affect the neurons

via the external input via a set of weights:  $I_i = gw_i E$ . In one of the experiments, we allowed for the proxy to the essential variable to modify the gain parameter instead:  $g_i = g_i + sw_i E$ , which is otherwise set to 1.

**Evolutionary algorithm** The parameters of the model were evolved using a real-valued genetic algorithm (Bäck, 1996). The optimization algorithm was run for populations of 100 individuals. We evolved the following parameters (ranges are shown in brackets):  $\tau_i$  [1, 10];  $w_{ji}$ ,  $\theta_j$ , and  $sw_i$  [-10, 10];  $gw_i$  [-0.5, 0.5]. Parameters were encoded in a vector of real-values between [-1, 1] and linearly mapped to their corresponding ranges. Each time the algorithm was run, individuals were initialized by random selection from the range of each parameter. Populations were seeded with center-crossing networks (Mathayomchan and Beer, 2002). At the end of a run, the parameters of the best performing individual were stored for later analysis.

**Fitness evaluation** Network activity is initialized to 0.0 once at the beginning of a trial. During a fitness trial, the environment changes to a new one every 200 units of time. The fitness of a trial is measured as the integral of the adapted variable over time.

$$f = \frac{\int_{t=0}^{200} w(t) (1 - a(t))}{c \int_{t=0}^{200} w(t)} \quad (2)$$

where  $t$  is time,  $a(t)$  is the binary adapted variable. The contribution of the adapted variable is weighted over time for each environment with a power function of time,  $w(t) = t^2$ , such that unadaptedness immediately after a change of environment weighs less than unadaptedness later on. Finally, in order to minimize the use of the critical boundary within the dynamics of stability, fitness is penalized for the number of times the agent becomes unadapted within a trial,  $c$ . The final fitness of a circuit is the average over all trials.

**Incremental evolution** The network is faced with two problems of a different nature. The first, to evolve attractors in each of the adapted regions, such that the network can stay within the adapted boundaries. The second, to transition between these regions. To allow for incremental evolution we divided the task into two stages. First, a minimal transition stage, where the network is asked to move from one environment to another one, always in the same sequence. Populations that are successful in this stage, move on to a second stage, where the network is asked to transition from any one environment to any other one, in random sequence.

**Alternative fitness evaluations** The fitness function defined above is based on the overt behavior, not the internal mechanisms of the circuit. In a final set of experiments, we use two fitness functions defined more mechanistically.



First, a function that maximizes the number of attractors in the phase-portrait. The network is started at different parts of the output space ( $3^C$  starting points that cover the full range of the constrained neurons) and integrated for 100 units of time. The fitness is calculated according to:  $f_a = (k/2^C)^5$ , where  $k$  are the number of attractors found in the circuit. The measure is normalized by the maximum number of attractors possible in the constrained circuit ( $2^C$ ). The proxy to the essential variable is fixed to 0 for this fitness evaluation. Second, a function that maximizes the area covered by a single dynamic in the phase-portrait. The network is started at one randomly chosen point in the output space and integrated for  $50 * 2^C$  units of time. The fitness is calculated by counting the number of corners of the constrained neural output space the circuit covers according to:  $f_b = (\sum_{i=1}^{2^C} m_i/2^C)^5$ , where  $m = 1$ , if the trace travelled near enough to the corner,  $d^* < 0.25$ , where  $d^*$  is the minimum distance to the corner in the full trajectory; otherwise  $m = 1.0 - ((d^* - 0.25)/0.75)$ . The proxy to the essential variable is fixed to 1 for this fitness evaluation. For this experiments, some of the ranges of the parameters were made larger:  $w_{ji}$ ,  $\theta_j$ , and  $sw_i$   $[-50, 50]$ , and  $w_{ii}$   $[4, 50]$ . When combined, the total fitness function was the product:  $f = f_a * f_b$ .

## Results

### 2D Ultrastability Task

**Two-neuron networks** The first set of experiments examined the ability of two-node CTRNNs to solve the simplest version of the ultrastability task: both neurons constrained (2C2N). In this task there are four possible environments, each with a respective internal viable region within the neural output space. The best two-node circuits attained a fitness of around 75%. That is, a typical successful network can only adapt to 3 of the four possible environments and transition between them in order (Figure 3). Evolutionary searches led to such networks 93 out of 100 times.

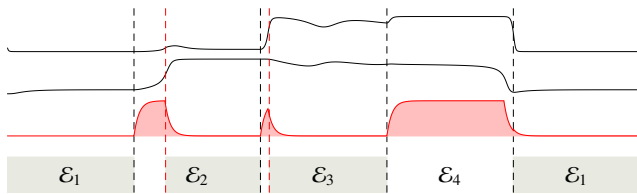


Figure 3: Neural traces during environment transitions for the 2C2N condition. Top two black traces represent neural output over time. Red trace represents the proxy to the essential variable. Black dashed vertical lines represent changes of environment. The environment is labeled at the bottom. Each environment is further divided by a red dashed vertical line into unadapted (no shade) and adapted (gray shade) stages.

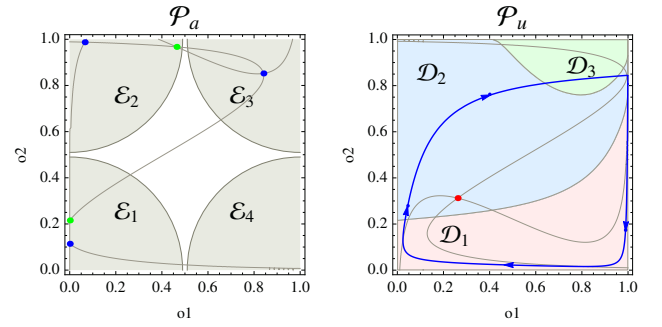


Figure 4: Phase portraits for the 2C2N condition. Adapted ( $\mathcal{P}_a$ ): stable equilibrium points (blue disks), saddle points (green disks), nullclines (solid gray curves), adapted regions for each environment (gray regions, labeled). Unadapted ( $\mathcal{P}_u$ ): unstable equilibrium point (red disk), limit cycle (blue trace). Basins of attraction of the adapted phase portrait are shown as colored volumes. The labels of the basins of attraction have a correspondence with the labels of the environments they are adapted to.

To understand the dynamics of the evolved circuit we analyzed the phase portraits of the most successful network when decoupled from the environment in the adapted and unadapted conditions independently by fixing the proxy to the essential variable to 0 or 1, respectively. The dynamics of a network are determined by the relative configuration of the nullclines. The network evolved three attractors (blue disks) in the phase portrait of the adapted condition ( $\mathcal{P}_a$ , Figure 4). The attractors cover three of the adapted regions (gray areas). In the unadapted condition, the network exhibits a limit cycle ( $\mathcal{P}_u$ , Figure 4). The basins of attraction, delimited by the saddle manifolds, of the attractors in the adapted phase portrait are visualized as light colored areas in the unadapted phase portrait. As can be seen, the limit cycle can transit between all evolved basins of attraction in the adapted phase. As we will see ahead, it is this combination of: (a) the shapes of the basins of attraction in the adapted phase portrait, and (b) the trajectory of the exploratory behavior in the unadapted phase portrait, that are crucial for the success of the adaptation.

To understand the limitations of the 2-neuron network, we need to understand how the proxy to the essential variable affects the dynamics of the network. A successful circuit needs two distinct types of dynamics: a rich repertoire of attractors, and a maximally exploratory dynamics. The maximum number of attractors in a circuit occurs when each nullcline intersects maximally. In the two node circuit, a limit cycle that can explore all the internal viable regions requires that the nullclines ‘fit’ inside of each other. There are other possible limit cycles in 2-neuron networks, but they don’t cover as much of the output space (Beer, 1995). The change in input to the neuron by the proxy to the essential variable



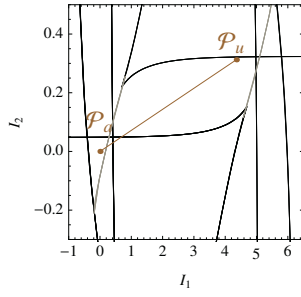


Figure 5: Parameter space for the 2C2N condition. Adapted phase ( $\mathcal{P}_a$ ). Unadapted phase ( $\mathcal{P}_u$ ). Regions of parameter that are crossed as the proxy to the essential variable,  $E$ , is gradually changed between 0 and 1 (brown line). Saddle node bifurcation (black curves). Hopf bifurcation (gray curves).

effectively shifts the nullcline up/down or left/right. A useful way to visualize this movement is in parameter space (Figure 5). The state of the proxy to the essential variable, by affecting the external input of the neurons, effectively moves the system through its parameter space. Different points in parameter space entail different dynamics. Crossing certain boundaries entails the system undergoing a bifurcation. Constrained to move in one direction in a line, the most successful circuits for this task evolve to navigate between a limit cycle and, at best, three stable attractor dynamics. From the limit cycle configuration, there is no way to shift both nullclines in one direction in a line to make them intersect maximally.

To understand the adaptive behavior of the coupled system, we need to understand the relationship between the shapes of the basins of attraction in the adapted phase portrait, and the trajectory of the exploratory behavior in the unadapted phase portrait. During a trial, the agent re-adapts to new environments by moving back and forth between the adapted ( $\mathcal{P}_a$ ) and unadapted ( $\mathcal{P}_u$ ) configurations. The timing of when the portraits change in relation to the basin boundary of the attractors is key to successful transitions. To illustrate this idea, we visualize the first transition of the trial shown in Figure 3, with the state of the network (brown trace) imposed over the changing phase-portrait (Figure 6). At the start of a run, the system starts off within the basin of attraction of the adapted environment ( $\mathcal{D}_1$ , Figure 4). The first time the environment changes, the proxy to the essential variable changes, the portrait of the system transitions towards the limit cycle configuration, and the state of the system begins to follow it ( $\mathcal{E}_2$ ,  $\mathcal{P}_u$ , Figure 6). When the system hits the boundary of the adapted region of the second environment, the proxy to the essential variable begins to return to normal, changing the portrait configuration back to the original ( $\mathcal{E}_2$ ,  $\mathcal{P}_a$ , Figure 6). Because the state of the system is now within the basin of attraction of the top-left

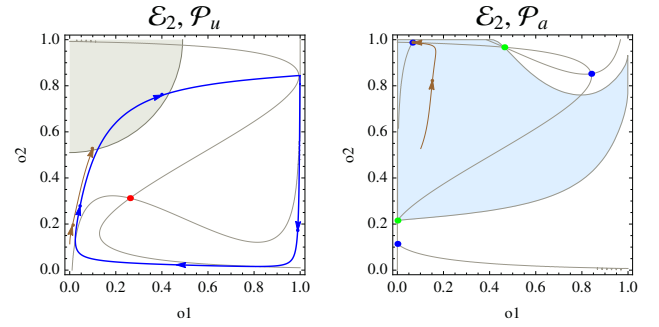


Figure 6: Neural activity (brown trace) in output space imposed over the phase portraits for the 2C2N condition. The two panels show the transition into  $\mathcal{E}_2$  (from  $\mathcal{E}_1$ ) from the example run in Figure 3. During the transition, the dynamics are first unadapted ( $\mathcal{P}_u$ ), then adapted ( $\mathcal{P}_a$ ).

attractor point, as soon as the limit cycle disappears, the system falls into it - effectively changing the dynamics of the system within the same configuration. A similar process occurs for each of the transitions.

The most successful 2-node networks displayed a similar dynamical configuration: (a) three equilibrium points in the adapted phase-portrait; (b) a limit cycle in the unadapted phase-portrait that allow them to transit between the basins of attraction of the adapted phase-portrait.

**Adding unconstrained neurons** One approach to obtain complete solutions to the 2-dimensional ultrastability problem is to increase the size of the network. This second set of experiments examined the ability of three-node CTRNNs to solve the same 2-dimensional version of the ultrastability task (2C3N). Same as before, there are four possible environments, each with a respective internal viable region within the neural output space of the two constrained neurons. The best three-node circuits attained a fitness close to 100%. Evolutionary searches led to successful networks (i.e., fitness greater than 98%) 38 out of 100 runs. A typical successful network was able to adapt to all four possible environments and transition between them in order (Figure 7). These successful networks can also transition successfully between any possible two regions. The networks use the output of the third unconstrained neuron (blue trace) to help them in both components of the task: (a) to have more attractors, and (b) to transition between them.

Similar to the two-neuron case, to understand the behavior of the network we can analyze the phase portraits of the dynamical system when decoupled from the environment in the adapted and unadapted conditions independently by fixing the proxy to the essential variable to 0 or 1, respectively (Figure 8). Analysis of a representative network shows four attractors in the phase portrait of the adapted condition (blue disks,  $\mathcal{P}_a$ ). The attractors cover each of the four adapted regions (gray volumes). In the unadapted exploratory condi-

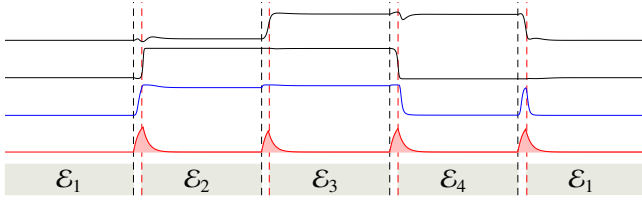


Figure 7: Neural traces for the 2C3N condition. Neural output of the constrained neurons over time (black traces). Neural output of the unconstrained neuron (blue trace). Proxy to the essential variable (red trace). Changes of environment (dashed vertical lines). Environment labeled at the bottom. Each environment is further divided by a red dashed vertical line into unadapted (no shade) and adapted (gray shade) stages.

tion, the network exhibits a limit cycle (blue trace,  $\mathcal{P}_u$ ). The limit cycle begins to form as soon as the proxy to the essential variable becomes greater than 0.0, which simultaneously turns on the third neuron (bifurcation diagram not shown). Crucially, for the purposes of the ultrastable behavior, the limit cycle transits through all of the basins of attraction of the adapted phase portrait (colored regions). During the exploratory phase, as soon as the state of the system becomes adapted with respect to the current environment, the proxy to the essential variable goes back to normal, the exploratory portrait gives way back to the multiple attractor portrait, and the state of the system is contained within the basin of attraction of the currently adaptive configuration.

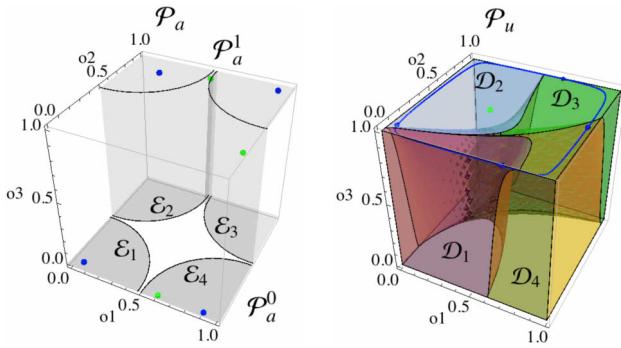


Figure 8: Phase portraits for the 2C3N condition. Adapted ( $\mathcal{P}_a$ ): equilibrium points (blue disks), saddle points (green disks), adapted regions for each environment (gray regions, labeled). The phase portraits can be divided into two depending on the output of the third neuron: off ( $\mathcal{P}_a^0$ ) or on ( $\mathcal{P}_a^1$ ). Unadapted ( $\mathcal{P}_u$ ): limit cycle (blue trace). Basins of attraction of the adapted phase portrait are shown as colored volumes. The labels of the basins of attraction have a correspondence with the labels of the environments they are adapted to.

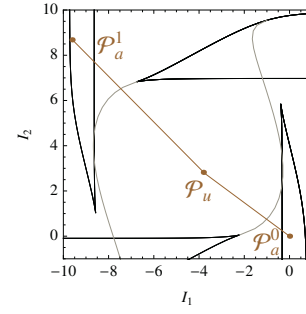


Figure 9: Parameter space for the 2C3N condition. Movement (brown trace) from the unadapted dynamics ( $\mathcal{P}_u$ ) to the adapted dynamics ( $\mathcal{P}_a^0$ ,  $\mathcal{P}_a^1$ ). The additional neuron allows for the system to affect the system in two different directions. Saddle node bifurcation (black curves). Hopf bifurcation (gray curves).

This network overcomes the limitation of the 2-neuron network using the added dimension from the additional unconstrained neuron. To see how, we can re-interpret this system as a two-neuron network with two parameters: the proxy to the essential variable and the third, unconstrained neuron. We can visualize the different dynamics of the network in this two dimensional parameter space (Figure 9). When the proxy to the essential variable is at rest, and the unconstrained neuron is off, the system is in a two-attractor configuration ( $\mathcal{P}_a^0$ ). Like in the two-dimensional case, when the proxy to the essential variable moves away from the base, the nullclines of both neurons shift in one direction in a line, moving the organization of the system towards a limit cycle configuration ( $\mathcal{P}_u$ ). From the same original configuration, leaving the proxy to the essential variable at rest, but activating the third neuron, moves the system in a different direction ( $\mathcal{P}_a^1$ ). Therefore, adding unconstrained components allows the nullclines of the constrained neurons to shift in multiple ways. Ultimately, from the limit cycle configuration, depending on the state of the third neuron, the system can transition to different regions of parameter space (brown lines) until it finds an internal viable region for the current environment.

**Modifying neuron gains** There are variations to the model that can overcome the limitations of the 2-neuron network. These variations are possible when the proxy to the essential variable can affect parameters other than the external input. This is an interesting condition to consider because recent work in neuroscience has shown how neuromodulators can modify the intrinsic properties of neurons (Marder and Thirumalai, 2002). For example, when the proxy to the essential variable is allowed to affect the gain of the circuit ( $g_i$  in Eq. 1), rather than the input, then it is possible to evolve a 2-neuron network that can adapt to all four environments, and transition between each of them. The

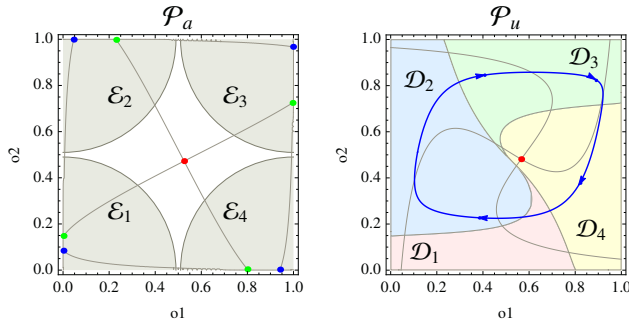


Figure 10: Phase portraits for the 2C2N where the proxy to the essential variable can affect the gains of the neurons. Adapted ( $\mathcal{P}_a$ ) and unadapted ( $\mathcal{P}_u$ ). The limit cycle in the unadapted phase-portrait transits all the basins of attraction of the adapted phase-portrait.

system easily changes between four attractors and a limit cycle (Figure 10). Change in the gain causes the nullclines to ‘twist.’ With this kind of modification available, it is possible for a system to change between the limit cycle configuration, where the nullclines are ‘fit’ inside each other, and the maximal-attractor configuration, where the nullclines intersect maximally, by ‘twisting’ the nullclines until they intersect the two other branches of the other nullclines.

### Ultrastability in Higher Dimensions

Evolving networks within the current restrictions of the task (i.e., internal viable regions in the corners of the neural output space) and model (i.e., proxy to the essential variable modifying only the external inputs) proved to be rather difficult for higher dimensions of the task. There are three tasks a behaviorally ultrastable system has to accomplish. First, it must maximize the number of attractors. Second, it must have access to a cyclic attractor that maximizes the coverage of its state space. Third, it must integrate the two dynamics so as to solve the ultrastable task. We designed a series of experiments to identify which of those tasks becomes harder in higher dimensions.

First, we tested each of the tasks individually, by directly evolving networks to either maximize the number of attractors or to maximize the area covered by network’s activity in output space. It was relatively easy to evolve larger networks to have the maximum number of attractors in their portraits. For example, after 5000 generations, 9 out of 10 circuits evolved successfully ( $> 99\%$  fitness) on the 5C5N task (i.e., 5-neuron network with 32 attractors). Similarly, it was easy to evolve larger networks to have dynamics that maximized the coverage of their output space. For example, after 5000 generations, 5 out of 10 circuits evolved successfully ( $> 99\%$  fitness) on the 5C5N task (i.e., a 5-neuron network with a single dynamic that allowed it to travel through all 32 corners of the neural output space). As far as we can

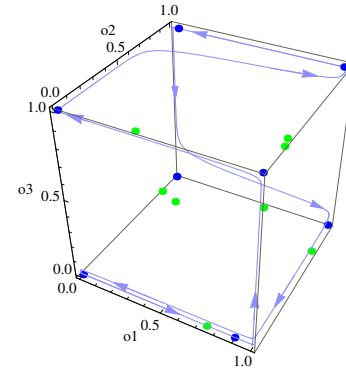


Figure 11: Phase portraits for the 3C5N condition. Projections in 3-dimensional space of the 5-dimensional phase portraits. Both adapted and unadapted portraits are shown in the same projection. The adapted portrait ( $\mathcal{P}_a$ ) contains 8 attractors in the corners (blue disks) and 8 saddle points (green disks). The unadapted portrait ( $\mathcal{P}_u$ ) contains one limit cycle (light blue trace) that transits close enough to all corners.

tell, the individual tasks scale well to higher dimensions.

We then tested evolving agents using the combined fitness function: one component to maximize the number of attractors while the proxy to the essential variable is set to 0 and another component to maximize the area covered by network’s activity in output space while the proxy to the essential variable is set to 1. We performed 50 evolutionary runs for 3C3N, 3C4N, and 3C5N tasks. We found no solutions for 3-, and 4-neuron networks. We found 4 out of 50 successful ( $>99\%$ ) 5-neuron networks. These results suggest the difficulty is in the integration of the two distinct behaviors.

As expected, the successful networks had 8 attractors in the adapted condition (blue disks, Figure 11), and an intricate cycle in the unadapted condition that passed relatively close to all the attractors (light blue trace, Figure 11). Due to the higher dimensionality, analyzing the basins of attraction of this network and their relationship to the limit cycle was not attempted. These networks were only partially successful because although they integrate the two tasks, when tested on the ultrastable task, they do not always transition successfully between all possible environments.

A number of key challenges remain to be studied. It will require further analysis to understand the difficulties encountered in higher dimensions. A few things to consider are: (1) adding unconstrained neurons; (2) further analyzing the dynamics of the partial solutions; and (3) using alternative arrangements of the internal viable regions.

### Discussion

This paper analyzes the dynamical possibilities for mechanisms of adaptation in living organisms based on Ashby’s framework of ultrastability. To explore alternative mecha-

nisms to the ones studied by Ashby and subsequent works, we developed evolutionary conditions that selected for behaviorally ultrastable systems. We analyzed the resulting neural networks using dynamical systems theory for the simplest conditions. The picture that emerged from our analysis opens up the idea of ultrastable mechanisms to a broader set than the original. It also allows for a more concrete instantiation of the framework within the context of neural networks.

The analysis in this paper was purposefully limited to the smallest networks in the lowest dimensional conditions. It's not hard to generalize our re-interpretation of ultrastable systems to higher dimensions. A system with many neurons can have a vast set of internal viable regions. The regions can form intricately connected networks in the high-dimensional output space. The proxies to the essential variables shift the rich repertoire of dynamics into the exploratory configuration, allowing the network to transition between different internal viable regions. The dynamics of exploration can be richer than just limit cycles, including for example chaotic dynamics. Re-adaptation corresponds to shifts in the phase portrait configuration: out of the exploratory regime, back into the rich dynamical repertoire. Crucially, in any one system, it is possible that only a fraction of the internal viable regions are explored during its lifetime, depending on the environmental changes experienced. In this view, the co-development of a rich dynamical repertoire together with a sufficiently encompassing exploratory dynamic occurs over evolutionary timescales, endowing agents with richer ultrastable systems to increase their probabilities of re-adaptation to novel environments.

Although the simplified 'internal viable regions' used in this study makes for tractable initial analysis, there are a number of conceptual variations not considered in this paper that are easily conceivable within the same framework. First, the shape and location of the region that is adaptive for any one environment can be any, not just the corners. Second, an internal viable region can contain more than one attractor, the boundaries of the region delimited by their combined basins of attraction. Third, internal viable regions for different environments can overlap. A simple re-interpretation of some of the networks analyzed in this paper could illustrate any of these variations. Interestingly, one idea that follows immediately from having richer internal viable regions is that which attractor within the region an agent finds itself in will depend on its history of interaction with the environment. Ultimately, the idea for this framework is to use internal viable regions defined by the actual interaction between an agent with certain essential requirements for its constitution and the environment.

The framework of behavioral ultrastability proposed in this paper could be developed further to study habit formation. In this paper, agents are evolved first to transition between environments in a specific order, and once successful, to transition between any possible environment. In both con-

ditions, the history of agent-environment interactions does not contain additional information about the likeliness of the next possible environment. An interesting space of solutions to study would be agents evolved to cope with a subset of all possible sequences of transitions. Changes from one sequence of transitions to a different sequence could involve a second level of adaptation, where the exploratory dynamics changes as a function of the sequence experienced. This would allow for the study of mechanisms with slower time constants that keep track of the sequence of transitions to modulate the exploratory dynamics.

## Acknowledgements

The revised paper benefited from discussions with Thomas Buhrmann, Matthew Egbert, and Inman Harvey. This work was supported by NSF grant No. IIC-1216739 and in part by TIN2011-24660. EUCogIII financed M.A.'s visit to I.U.

## References

- Ashby, W. (1960). *Design for a brain: The origin of adaptive behavior*. Chapman & Hall, 2nd edition.
- Bäck, T. (1996). *Evolutionary algorithms in theory and practice*. Oxford University Press.
- Beer, R. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3:471–511.
- Di Paolo, E. A. (2000). Homeostatic adaptation to inversion in the visual field and other sensorimotor disruptions. In Meyer, J.-A., Berthoz, A., Floreano, D., Roitblat, H., and Wilson, S., editors, *From Animals to Animats 6*, pages 440–449. MIT Press.
- Harvey, I. (2008). Homeostasis via chaos: implementing the uniselectors as a dynamical system. In Bullock, S., Noble, J., Watson, R. A., and Bedau, M. A., editors, *Proc. of the Eleventh Int. Conf. on Artificial Life*, page 774. MIT Press.
- Herrmann, J., Hilicki, M., and Der, R. (2004). On Ashby's homeostat: A formal model of adaptive regulation. In Schaal, S., Ijspeert, A., A., B., Vijayakumar, S., Hallam, S., and Meyer, J.-A., editors, *From Animals to Animats 8*, pages 324–333. MIT Press.
- Iizuka, H. and Di Paolo, E. A. (2008). Extended homeostatic adaptation: Improving the link between internal and behavioural stability. In Asada, M., Hallam, J., Meyer, J.-A., and Tani, J., editors, *From Animats to Animals 10*, pages 1–11. Springer.
- Izquierdo, E., Harvey, I., and Beer, R. (2008). Associative learning on a continuum in evolved dynamical neural networks. *Adaptive Behavior*, 16:361–384.
- Marder, E. and Thirumalai, V. (2002). Cellular, synaptic and network effects of neuromodulation. *Neural Networks*, 15:479–493.
- Mathayomchan, B. and Beer, R. (2002). Center-crossing recurrent neural networks for the evolution of rhythmic behavior. *Neural Computation*, 14:2043–2051.
- Yamauchi, B. and Beer, R. (1994). Sequential behavior and learning in evolved dynamical neural networks. *Adaptive Behavior*, 2:219–246.



# The influence of assortativity on the robustness and evolvability of gene regulatory networks upon gene birth\*

Dov A. Pechenick<sup>1</sup>, Jason H. Moore<sup>1</sup>, and Joshua L. Payne<sup>2</sup>

<sup>1</sup>Computational Genetics Laboratory, Dartmouth College, Hanover, New Hampshire, USA

<sup>2</sup>Institute of Evolutionary Biology and Environmental Studies, University of Zurich, Zurich, Switzerland  
Dov.A.Pechenick.GR@Dartmouth.edu

## Abstract

Gene regulatory networks (GRNs) represent the interactions between genes and gene products, which drive the gene expression patterns that produce cellular phenotypes. GRNs display a number of characteristics that are beneficial for the development and evolution of organisms. For example, they are often robust to genetic perturbation, such as mutations in regulatory regions or loss of gene function. Simultaneously, GRNs are often evolvable as these genetic perturbations are occasionally exploited to innovate novel regulatory programs. Several topological properties, such as degree distribution, are known to influence the robustness and evolvability of GRNs. Assortativity, which measures the propensity of nodes of similar connectivity to connect to one another, is a separate topological property that has recently been shown to influence the robustness of GRNs to point mutations in *cis*-regulatory regions. However, it remains to be seen how assortativity may influence the robustness and evolvability of GRNs to other forms of genetic perturbation, such as gene birth via duplication or *de novo* origination. This abstract outlines a recent publication, in which we employed a computational model of genetic regulation to investigate whether the assortativity of a GRN influences its robustness and evolvability upon gene birth. We considered GRNs to be robust if they conserved all their phenotypes (attractors) following the introduction of a new gene, and evolvable if they were able to innovate at least one novel phenotype. We found that the robustness of a GRN generally increases with increasing assortativity, while its evolvability generally decreases (Figure 1; above), and this results in an increased proportion of assortative GRNs that are simultaneously robust and evolvable (Figure 1; below). This is due to: (1) Assortative GRNs have shorter attractors, which are more likely to be conserved (Figure 2), and (2) assortative GRNs have smaller out-components, resulting in a reduced chance of innovation (Figure 3). This work extends our understanding of how the assortativity of a GRN influences its robustness and evolvability to genetic perturbation.

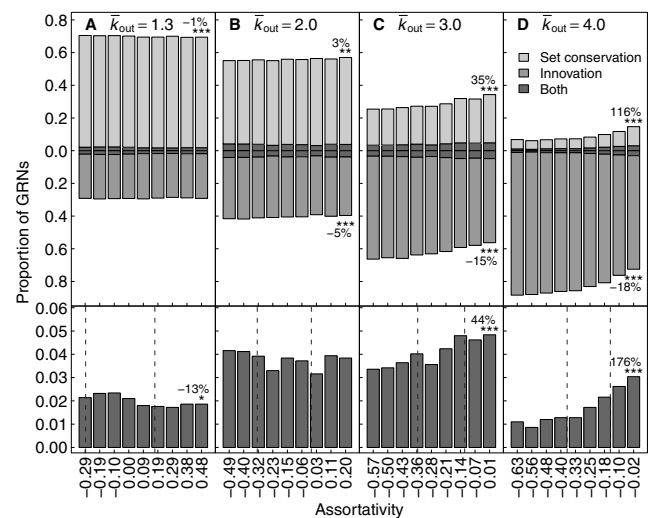


Figure 1: Proportion of GRNs with set conservation, innovation, or both as a function of assortativity. (Above) Light gray bars show the proportion of GRNs at a fixed assortativity value that exhibited set conservation after gene birth. Medium gray bars represent the proportion that exhibited innovation. Dark gray bars show the overlap, which is the proportion that both conserved and innovated. Each bar is a proportion of 5000 GRNs for each of:  $\bar{k}_{out} \in \{1.3, 2.0, 3.0, 4.0\}$  and 9 assortativity values. (Below) The proportion of the overlap is shown at a zoomed-in scale. Note that total proportions of conservation and innovation may not add up to 1, as some GRNs may exhibit neither. The percentage change from the smallest to the largest assortativity value is shown for conservation, innovation, and both. Statistical significance for Spearman's correlation is denoted by \* ( $p < 0.05$ ), \*\* ( $p < 0.01$ ), or \*\*\* ( $p < 0.001$ ). Vertical dashed lines show the minimum and maximum assortativity values for the middle 95% of the null distribution for each  $\bar{k}_{out}$ .

\* Published in Journal of Theoretical Biology, 2013, <http://dx.doi.org/10.1016/j.jtbi.2013.03.019>



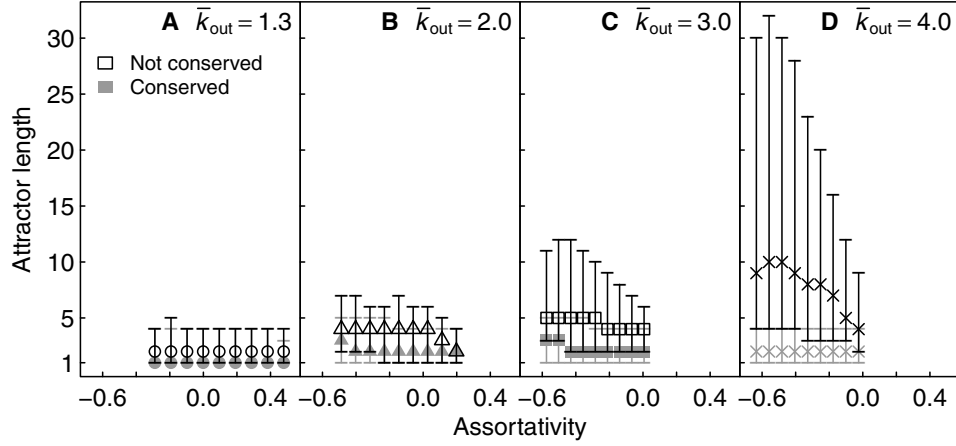


Figure 2: Lengths of attractors that were or were not conserved as a function of assortativity. Each point represents the median length of unique attractors before gene birth in 5000 GRNs at a fixed assortativity value. Error bars represent the 25<sup>th</sup> and 75<sup>th</sup> percentiles. Medians and percentiles for attractors that were not conserved are shown in black, whereas those for attractors that were conserved are shown in gray. GRNs are grouped according to their  $\bar{k}_{out}$  and assortativity value, as in Figure 1. Conserved attractors are significantly shorter for every assortativity value and every  $\bar{k}_{out}$  (Wilcoxon Rank Sum test,  $p \ll 0.001$ ).

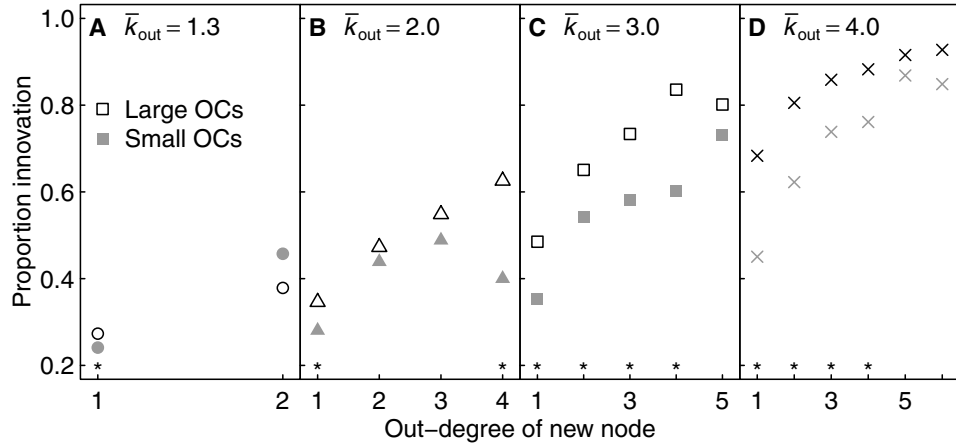


Figure 3: Proportion of innovation as a function of the out-degree of the newly introduced node, and whether the new node possesses a large or small out-component (OC). The OC of a node  $i$  in the GRN is the set of nodes in the GRN that is reachable via directed paths starting from  $i$ . The 5000 GRNs at the highest assortativity value for each  $\bar{k}_{out}$  were first binned by the out-degree of the new node, since this property positively influences the ability of the new node to cause innovation. Each out-degree bin was then split into two groups according to whether the new node possesses a large OC (larger or equal to the median OC size) or small OC (smaller than the median OC size). Black markers represent the proportion of GRNs that innovated at least one attractor where new nodes possess large OCs, and gray markers represent innovation for GRNs where new nodes possess small OCs. Asterisks mark significant differences in proportions between large and small OC categories ( $p < 0.05$ , Pearson's chi-squared test). Only out-degrees for which at least 30 GRNs were present in large and small OC bins are plotted.

## Self-organized game dynamics in complex networks

Flávio L. Pinheiro<sup>1,2,3</sup>, Vítor Vasconcelos<sup>1,2,3</sup>, Francisco C. Santos<sup>3,1</sup> and Jorge M. Pacheco<sup>4,5,1</sup>

<sup>1</sup>ATP-group, CMAF, Instituto para a Investigação Interdisciplinar, P-1649-003 Lisboa Codex, Portugal

<sup>2</sup>Centro de Física da Universidade do Minho, 4710 - 057 Braga, Portugal

<sup>3</sup>INESC-ID & Instituto Superior Técnico, IST-Taguspark, 2744-016 Porto Salvo, Portugal

<sup>4</sup>Departamento de Matemática e Aplicações da Universidade do Minho, 4710 - 057 Braga, Portugal

<sup>5</sup>Centro de Biologia Molecular e Ambiental da Universidade do Minho, 4710 - 057 Braga, Portugal  
flavio.lpp@gmail.com

Complex networks are ubiquitous and known to profoundly affect the processes that take place on them. From a theoretical perspective, some of the most complex processes studied to date, occurring on complex networks, are related with behavioural dynamics and decision-making, often described by means of social dilemmas of cooperation. Among these, the Prisoner's Dilemma (**PD**) provides the most popular metaphor of such dilemmas, given that its only Nash equilibrium is mutual defection, despite mutual cooperation providing higher returns – thus the dilemma. We may also assume a population dynamics (evolutionary) approach to game theory where agents revise their behaviour based on the perceived success of others, creating a gradient of selection which dictates how cooperation self-organizes through time. Evolutionary Games provide one of the most sophisticated examples of complex dynamics in which the role of the underlying network topology proves ubiquitous. For instance, when cooperation is modeled as a prisoner's dilemma game, cooperation may emerge (or not) depending on how the population is networked (Santos et al., 2012a).

Up to now, it has been hard to characterize in detail the global dynamics by which local self-regarding actions lead to a collective cooperative scenario while relating it to the network topology. Indeed, most network studies have been focused on the analysis of the evolutionary outcome of cooperation – either by means of the numerical analysis of steady states or by the analytical determination of the conditions that lead to fixation – without characterizing the self-organization process by which one of the strategies outcompetes the other. Here we report on new results (Pinheiro et al., 2012a,b; Santos et al., 2012b,a), where we show how to establish the link between individual and collective behavior in the context of evolutionary games in networked populations.

Overall, we show how behavioral dynamics of individuals facing a cooperation dilemma in social networks can be understood as though individuals face a different dilemma in absence of structure. As illustrated in Fig. 1, homogeneous networks promote a coexistence dynamics between cooperators and defectors – akin to the Chicken or Snow-

drift game – whereas heterogeneous networks, from single to scale-free networks, favor the coordination between them, similar to the Stag-hunt game. In other words, while agents locally perceive and play a **PD**, globally the dynamics of the population resembles the one obtained from a completely different game, as if, individuals would be locally facing a different dilemma.

To this end we define a time-dependent variable – that we call the average gradient of selection (**AGoS**) – and use it to track the self-organization of cooperators when co-evolving with defectors. In finite well-mixed populations, that is, populations in which every individual can interact with every other individual in the population, this gradient of selection ( $G(x_C) = T^+(x_C) - T^-(x_C)$ ) can be computed analytically, as the difference of the probabilities of increasing ( $T^+(x_C)$ ) and decreasing ( $T^-(x_C)$ ) the number of cooperators, for each fraction  $x_C$  of cooperators. While this quantity is impossible to be attained analytically for arbitrary network structures, the **AGoS** provides a numerical account of the same variables, offering the change in time of the frequency of cooperative traits under selection. The **AGoS** can be computed for arbitrary intensity of selection, arbitrary population structure and arbitrary game parameterization. We further prove that the global games are not fixed: they change in time, co-evolving with the motifs of cooperators in the population. The evolutionary outcome of such a self-organization process will depend sensitively on this co-evolution, which can be followed using a time-dependent **AGoS**.

The scenarios illustrated in Fig. 1 become even richer whenever one takes into account the role of selection pressure (also known as intensity of selection) in the overall evolutionary dynamics of a networked population. Selection pressure provides the relative significance of agent fitness in the evolutionary process, as opposed to an arbitrary or random adoption of strategies. This is important, as selection pressure can be very different depending on the processes at stake. Indeed, in many social interactions, errors in decision making, perhaps induced by stress or exogenous confounding factors, which often translate into a bounded rational be-

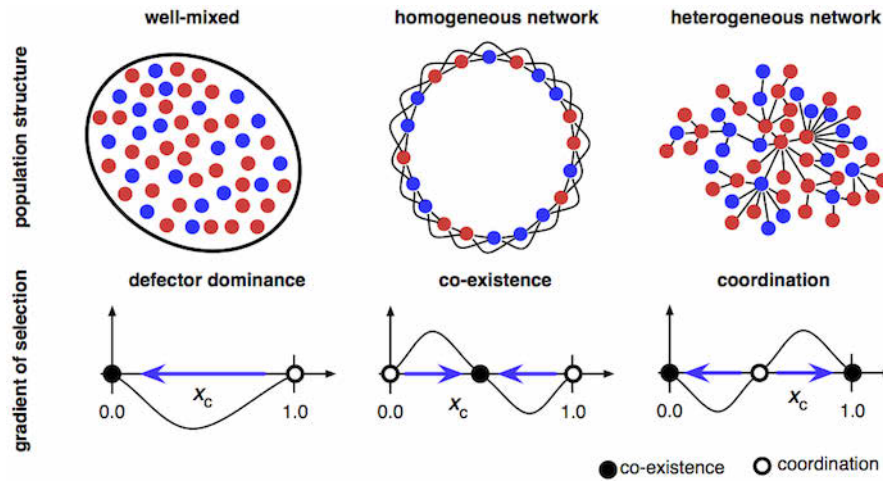


Figure 1: The Average Gradient of Selection (AGoS) provides a characterization of the change in time of the fraction of cooperators under natural selection, being positive (negative) when the fraction of cooperators tends to increase (decrease). While in well-mixed populations (left panel), the tragedy of the commons ( $x_c = 0$ ) emerges as the only stable fixed point, homogeneous networks favor the co-existence of cooperators and defectors (middle panel), whereas degree heterogeneous networks (right panel) creates two basins of attraction, as if agents would be locally facing a coordination dilemma.

haviour of the players, may lead to an overall weak selection environment. This contrasts with many other situations where selection is mainly strong such as the dynamics of cultural evolution. Moreover, the fate of cooperation in social networks may depend on how the success of the others is locally perceived, which is related with the number of partners of each player and their social context, turning selection pressure into a central variable in behavioural evolution.

As we show in (Pinheiro et al., 2012b), as opposed to what happens in heterogeneous populations that generate a coordination dynamics for a broad range of selection pressure values, on homogeneous networks the population-wide dynamics depends on the intensity of selection: under strong selection they favour a co-existence like dynamics while under weak selection we recover the *well-mixed* scenario of a **PD**-like dynamics which leads to the demise of cooperation (Fig. 1, left panel). Additionally, we were able to identify the existence (on several types of networks) of an optimum level of selection pressure for which cooperation is maximised. The underlying process that leads to this result differs from homogeneous to heterogeneous networks. In the first class of networks the optimal selection pressure is associated with the ability of cooperators to form and sustain clusters, while on the second class it is the result of a decoupling in the distribution of intensities of selection between pairs of agents that is present from the natural diversity of fitnesses (Santos et al., 2012a) in the population.

Additionally, the application of the AGoS is not limited to 2-person games. In fact, as discussed in (Santos et al., 2012b), heterogeneous network structures create multiple internal equilibria when individuals face public goods

dilemmas, departing significantly from the reference scenario of a structureless population, approach which can also be extended to N-person games in adaptive networks (Moreira et al., 2013). Finally, we would like to stress that the scope and importance of this methodology goes far beyond the present application to evolutionary games on graphs. The principles can be used to extract any dynamical quantity that describes a process (as long as it is a Markov process) taking place on a network such as the outbreak of epidemics or the spread of information.

**Acknowledgements.** This research was supported by FCT-Portugal (refs. SFRH/BD/77389/2011, SFRH/BD/86465/2012, PTDC/FIS/101248/2008, PTDC/MAT/122897/2010 and PEst-OE/EEI/LA0021/2011).

## References

- Pinheiro, F. L., Pacheco, J. M., and Santos, F. C. (2012a). From local to global dilemmas in social networks. *PLoS One*, 7(2):e32114.
- Pinheiro, F. L., Santos, F. C., and Pacheco, J. M. (2012b). How selection pressure changes the nature of social dilemmas in structured populations. *New J. Phys.*, 14(7):073035.
- Santos, F. C., Pinheiro, F. L., Lenaerts, T., and Pacheco, J. M. (2012a). The role of diversity in the evolution of cooperation. *J. Theor. Biol.*, 299:88–96.
- Santos, M. D., Pinheiro, F. L., Santos, F. C., and Pacheco, J. M. (2012b). Dynamics of n-person snowdrift games in structured populations. *J. Theor. Biol.*, 315:81–86.
- Moreira, J. A., Pacheco, J. M., and Santos, F. C. (2013). Evolution of collective action in adaptive social structures. *Sci. Rep.*, 3:1521.

# Evolution of Incremental Complex Behavior on Cellular Machines

Stefano Nichele<sup>1</sup> and Gunnar Tufte<sup>1</sup>

<sup>1</sup>Norwegian University of Science and Technology, Department of Computer and Information Science,  
Sem Selandsvei 7-9, 7491, Trondheim, Norway  
{nichele, gunnar}@idi.ntnu.no

## Abstract

Complex multi-cellular organisms are the result of evolution over billions of years. Their ability to reproduce and survive through adaptation to selection pressure did not happen suddenly; it required gradual genome evolution that eventually led to an increased emergent complexity. In this paper we investigate the emergence of complexity in cellular machines, using two different evolutionary strategies. The first approach is a conventional genetic algorithm, where the target is the maximum complexity. This is compared to an incremental approach, where complexity is gradually evolved. We show that an incremental methodology could be better suited to help evolution to discover complex emergent behaviors. We also propose the usage of a genome parameter to detect the behavioral regime. The parameter may indicate if the evolving genomes are likely to be able to achieve more complex behaviors, giving information on the evolvability of the system. The experimental model used herein is based on 2-dimensional cellular automata. We show that the incremental approach is promising when evolution targets an increase of complexity.

## Introduction

Evolved artificial developmental systems' remarkable range of products, i.e. biological organisms, with variety in form, function and "complexity", all tailored to fill their niche, is an alluring design concept. Such bio-inspired design methodology can be used for any kind of system to create a variety of artifacts that can handle different problems.

However, knowledge and methods to be able to exploit similar core processes [14] for the design of artificial organisms are still subjects of exploration and research.

Evolved artificial developmental systems have shown many favorable features that are borrowed from natural biological systems, such as the main subject here, the ability to evolve inherent complexity as a response to evolutionary pressure [27]. Evolvable, or in particular EvoDevo systems are products of bottom-up processes, in contrast to typical top-down engineering design approaches. A system emerging as a product of a bottom-up process can target system properties out-of-bounds for traditional top-down designed artifacts. Self-organization, self-construction, adaptivity, scalability and robustness are all example of such hard to reach properties.

In contrast to the open ended evolution in nature, evolution of artificial EvoDevo systems often includes an expressed goal; fitness is a kind of usability [15] measurement. The target functionality is defined and thus placed within some

complexity measure. Such complexity can be defined at several levels. The complexity of the machine's composition, i.e. the number of components and connections can be quantified. Another complexity measurement may be functionality in terms of information processing. Quantification of complexity in artifacts and biological organism has no common defined unit of measurement or ratio of comparability. However, intuitively there are differences. Such differences can be related to the composition of artifacts/organisms or as a measure of their functionality. If complexity, for an organism, is a measurement of functional properties within its environmental niche, different levels of behavioral complexity can be said to exist. In this context, high complexity is not a goal in itself; it is merely a product of the species adaptation to be able to reproduce. The genetic information included in the genotype and the developmental processes for any particular specie has evolved and diverged through adaptation from the primordial soup. As such, the behavioral complexity of organisms is a product of the evolved interplay between genetic information and developmental processes.

As a step toward more knowledge of underlying processes and finding design methodologies for the exploitation of EvoDevo for artificial systems, we investigate how a gradual change in the complexity requirements, in evolutionary time, influence on EvoDevo system's ability to evolve complexity. Further, a variation of Langton's Lambda parameter [16] is used as an indicator of evolutionary genome adaptation to resulting phenotypic complexity. The taken experimental approach uses a kind of incremental complexity evolution to simulate the process of species adaptation to a changing environment requiring growth of complexity. A 2D cellular developmental model based on Cellular Automata (CA) is used in the experiments, so as to be able to visualize artificial organisms' development in 2 dimensions.

In our incremental evolutionary approach, the evolution process tackles the problem of targeting complexity incrementally. Instead of seeking for maximum complexity in the early generations, the problem is divided in sub-problems. Generations are divided in intervals and in every interval the target complexity demand is increased, keeping the target functionality unvaried, i.e. the class of problems is the same. In such way, it may be possible to evolve favorable genes for intermediate complexity levels, which may be beneficial in order to achieve higher complexity in the long term.

Since no universally accepted definition of complexity exists, many authors use it implicitly without specifying which



notion of complexity they are using. Yet, without any common measure of genotype or phenotype complexity, any significant claim is not verifiable. Genome complexity measures may sometimes be unfitting, since the amount of information encoded in the genome is not directly proportional to the complexity of the emergent organisms. Even in nature, some unicellular eukaryotic organisms have much larger genomes than humans. In addition, there are other factors that impact on the organisms' complexity during their growth, e.g. the environment.

One may argue that complexity of an EvoDevo system may be measured in terms of information contained in a genome, by ranking through a Turing machine, by quantifying the capacity for a genome to exploit a provided area of growth, using approximations of Kolmogorov complexity [5, 6, 9]. The used complexity measure is based on compression of CA behavior in terms of trajectory and attractor lengths, a kind of adaptation of principles from Kolmogorov complexity.

The article is laid out as follows: background information and motivation on incremental evolution is presented in Section 2. In Section 3 Lambda genome parameter is introduced and in Section 4 the developmental model used in the experiments is described. Section 5 explains the genetic algorithms used herein. The experimental setup is illustrated in Section 6 and in Section 7 the results of the experiments are presented together with a discussion of the ideas and the results. Finally Section 8 concludes the work.

## Incremental Evolution

A general evolutionary strategy may be too difficult for the evolution system to discover possible solutions directly. Instead, it is possible to learn complex behaviors incrementally, starting from a simple behavior and gradually making the task more challenging [1]. Incremental evolutionary approaches have been used successfully to evolve complex behaviors step by step. Many studies investigated the training of artificial neural networks with Genetic Algorithms (GAs), in order to evolve robots controller able to perform complex action sequences, e.g. complex light switching behavior [2] or robot duels controllers [3]. This approach has shown interesting results, being able to evolve converged populations to the new task. On the other hand, conventional evolutionary methods may have too high selective pressure in the early stages of the evolution, getting the GA blocked in an unfruitful area of the search space. If the population is first evolved to an easier behavior, it may be possible to discover and access a region of the solution space where even more complex behaviors are more likely to be found. As such, the ultimate complex behavior may be reached incrementally by evolving a sequence of intermediate behaviors with growing complexity:

Behavior 1  $\xrightarrow{\Delta}$  B 2  $\xrightarrow{\Delta}$  ....  $\xrightarrow{\Delta}$  B n-1  $\xrightarrow{\Delta}$  B n

In this way, genotypes are evolved gradually and the search is driven on solutions that are likely to benefit and retain existing capabilities. Conventional evolution tends to fluctuate between idiosyncratic but still not interesting solutions [12]. Incremental evolution may foster continuing innovation by elaborating on available solutions.

## Genome Parameter: $\lambda$

In our cellular developmental model, we are aiming to target complex phenotypic properties. Attractor length, i.e. development reaches a structure or state that is stable by self-regulation (point attractor) or a dynamic phenotypic structure that is self-reorganizing (cyclic attractor), is the chosen metric. The strategy is therefore to evolve intermediate genotypes that develop and express specific attractor lengths. Every fixed number of generations, we increase the sought attractor length value to increment the complexity demand.

In terms of evolvability, since we want to investigate if the evolving genotypes are able to evolve and develop more complex phenotypes, we attempt to measure the behavioral regime using a genome parameter. Parameters obtained from the genome information can be used to estimate the dynamic behavior of the system. In this work, the genotypes are represented as a transition rule table, where developmental actions are defined as a function of the neighborhood configuration (see next chapter for details). In this way, it is possible to analyze the different developmental actions and calculate parameters obtained from the genome table.

Several genome parameters have been previously proposed in order to measure genotype properties. Langton [16] studied a parameter  $\lambda$  as a measure of the activity level of the system and its disorder. A similar parameter, neighborhood dependent, is Absolute Activity presented by de Oliveira [20]. Li [21] introduced Mean Field Parameters to monitor if the majority of the regulatory actions follow the “mean” configuration. de Oliveira [20] presented a very similar parameter called Neighborhood Dominance. Binder [22, 23] introduced the Sensitivity parameter which measures the number of changes in the output of the transition table based on a change in the neighborhood, one cell at a time, over all the possible neighborhoods of the rule being considered. This has also being studied by de Oliveira [24, 20] as Context Dependence. Different properties of genome parameters have been investigated in details in [11]. In particular, the  $\lambda$  parameter has shown interesting abilities to discriminate genotypes in different behavioral classes, e.g. fixed, chaotic, random [13]. As such, we monitor the  $\lambda$  value along the evolutionary process.  $\lambda$  is calculated according to Equation 1.

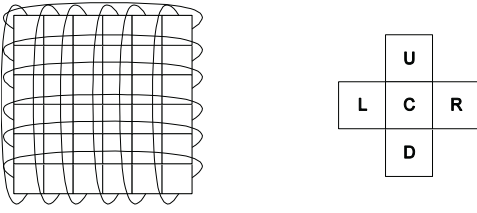
$$\lambda = \frac{K^N - n}{K^N} \quad (1)$$

$n$  represents the number of transitions to the quiescent state (i.e. inactive or dead state),  $K$  is the number of cells types and  $N$  is the neighborhood size (see following section for details).

## Cellular Developmental Model

The developmental model used in this work is a minimalistic cellular developmental model based on cellular automata, similar to cellular models used in [25, 26, 19]. The system herein is close to the field of Morphogenetic Engineering [7], where the goal is “self-architecturing” systems. In embryomorph systems [8], the approach is based on embryogenesis: the self-assembly of myriads of cells starting from a single zygote which holds the complete genotype information.





**Fig. 1:** Developmental model with cyclic boundary conditions and von Neumann neighborhood configuration.

A CA can be considered as a developing organism, where the genome specifications and the gene regulation information control the cells' growth and differentiation. The behavior of the CA is then represented by the emerging phenotype, which is subject to size and shape modifications, according to the cellular changes along the developmental process. Such dynamic developmental system can show adaptation, self-modification, plasticity [18] or self-replication properties [17].

L	R	U	D	C	$C_{(t-1)}$
0	0	0	0	0	0
0	0	0	0	1	{0,1,2}
0	0	0	1	0	{0,1,2}
0	0	0	1	1	{0,1,2}
0	0	1	0	0	{0,1,2}
		:			:
1	1	1	1	1	{0,1,2}
0	0	0	0	2	{0,1,2}
0	0	0	2	0	{0,1,2}
0	0	0	2	1	{0,1,2}
0	0	0	2	2	{0,1,2}
		:			:
2	2	2	2	2	{0,1,2}

**Fig. 2:** Developmental table with neighborhood configurations and relative developmental actions.

The model is based on a two-dimensional cellular automaton with cyclic boundary conditions, as shown in Figure 1. The number of cell types is set to three (type 1 and 2 plus the quiescent or dead cell type 0) in order to keep the property of multicellularity. A single cell (zygote) is placed in the centre of the development grid and develops according to a developmental table based on von Neumann's neighborhood (five neighbors). All the possible regulatory input combinations are explicitly represented in the development table, i.e. 243 ( $3^5$ ) neighborhood configurations.

To ensure that cells will not materialize where there are no other cells around, a restriction has been set: if all the neighbors of an empty cell are empty, the cell will be empty also in the following development step. This is represented in the first entry of the developmental table in Figure 2. A more detailed description of the development model is given in [10, 11].

## Genetic Algorithm

The Genetic Algorithm used in the experiments is tested with two different fitness functions: a classical fitness approach that

targets the maximum complexity and an incremental growth of fitness. An incrementally growing fitness denotes a changing environment that requires more complex behaviors to survive. It must be underlined that the two different fitness functions could in theory perform the same way. The same genotypes could be discovered through evolution since there are no restrictions in the areas of the search space that are being explored. Anyway, this is very unlikely to happen, since the environment is interpreted differently on the evolutionary time scale. The GA consists of a population of ten individuals and uses a roulette wheel technique for proportionate selection of two potentially useful individuals. The worst three elements are replaced by two new individuals that are copies of the two selected ones with mutation rate 0.02 for each of the entries in the developmental table. The third new element is generated by uniform one-point crossover of the two selected individuals.

As we mentioned, the main difference between the two evolutionary strategies lies in the fitness function. In the classical scenario, the fittest individual is the one with longest attractor length. In the incremental approach the fittest is the individual with smallest difference between the actual length and the target length in that specific generation, i.e. environmental requirements at specific moments in evolutionary time.

The target length is defined as follows: in the first generation it is set as 1, i.e. point attractor. In this phase, the GA searches for phenotypes that end up with a single point attractor. Every fixed number of generations, the target value is incremented by a constant value. It is expected that in the following interval of generations, the population will be able to evolve and adapt towards the new target, i.e. an increasing complexity demand for longer attractor length along the evolutionary timeline.

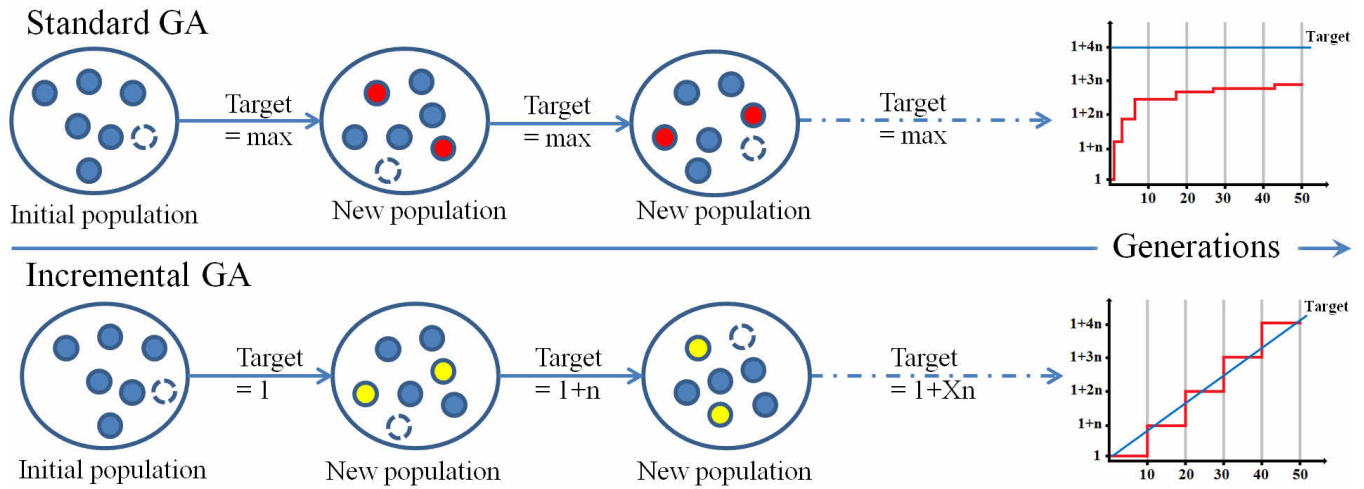
Details on the development process, number of generations, length of the intervals and initial conditions of the genotypes are given in the next section, which describes the experimental setup. Source code is available upon request.

## Experimental Setup

In the experiments herein, the main idea is to generate an initial population of ten genotypes, develop the corresponding phenotypes (starting from a single cell placed in the centre of the grid) until an attractor is found, evaluate the phenotypes with a fitness function and evolve the chosen genomes throughout the generations. This process is repeated for GA with standard fitness function and GA with incremental fitness function. The performances of the different algorithms are evaluated, measuring the ability to achieve sought complexity in terms of attractor length, i.e. number of development steps between two repetitions of the same state. This experimental setup is represented graphically in Figure 3.

Two different strategies of generating the initial population are investigated:

- From “dead genomes”: all the transitions in the developmental table lead to the dead state and the value of  $\lambda$  is uniform. In this scenario, the GA has to evolve “dead” genomes, i.e. the developed phenotype results in a dead



**Fig. 3:** Experimental Setup: GA with standard fitness on top (targeting maximum complexity), GA with incremental fitness on bottom (target fitness increased gradually)

organism after the first development step, towards “alive” genomes. This approach could be interesting since favorable genes are evolved from scratch, especially when random initialization is not possible or feasible.

- From random genomes: the initial population of genomes is initialized randomly<sup>1</sup>. In this way, it is possible to have extremely fit genomes, or very unfit, from the first generations. In both cases, it may be particularly difficult to evolve towards genomes with favorable characteristics to reach the defined goal. The parameter  $\lambda$  has a distributed value.

The way genomes are generated has a strong impact on how the resulting phenotypes will behave. Trying to understand the relation between genotypes and possible resulting phenotypes means understanding which kind of information is present in the genome and which behavioral properties may emerge. Since in our model all the possible regulatory combinations are fully specified together with the corresponding developmental actions, it is possible to calculate the Lambda genome parameter for each individual in the population.  $\lambda$  is calculated out of the regulative outcome in the developmental table, i.e. column  $C(t+1)$  in Figure 2. Following Langton’s definition [16], a quiescent state must be chosen. We choose the void cell type (type 0) as the quiescent state. Lambda is then calculated by  $1 - \text{the ratio between transitions to the quiescent state and the total number of transitions in the developmental table}$ . It is implicit that if the population is initialized with dead genomes, all the transitions in the developmental table will lead to the quiescent state. Thus,  $\lambda$  will be 0, which means genotypes with low behavioral activity. On the other hand, when the population is initialized with random genomes,  $\lambda$  is more likely in the vicinity of a critical behavioral regime, near the Edge of Chaos [16]. In this area of the solution space, it is more likely to find complex behaviors.

<sup>1</sup> Marsenne twister is used for initialization of randomized genotypes and genetic operators (mutation and crossover).

Monitoring Lambda along all the evolutionary process will give information on the ability of the population to evolve and adapt to the target complexity level.  $\lambda$ , as an indication of computation, has been discussed in [28]. However, from previous work [16, 11], we know that the attractor length of a certain organism is strongly related to its  $\lambda$  value, which can be calculated from the genome composition. As such, Lambda could be used to drive evolution in desired parts of the search space where the desired behavior is more likely to be found. This is part of ongoing experimentation.  $\lambda$  here is measured to gain information of the evolution of genome composition and interpreted as an indicator of the evolvability of the system, which may confirm our hypothesis that an increase in complexity is more likely to happen if evolutionary search leads towards the desired behavioral regime.

## Results and Discussion

In the experiments herein, the array size of the CA was set to  $4 \times 4$ . The size of the arrays was chosen as to be able to carry out experiments in reasonable computational time.

Organisms of  $4 \times 4$  cells may be considered rather small; however, the theoretical maximum attractor length is  $3^{16}$ . As such, even at the chosen array size, the number of development steps to reach an attractor could be rather big.

### Experiment 1: Dead Genomes

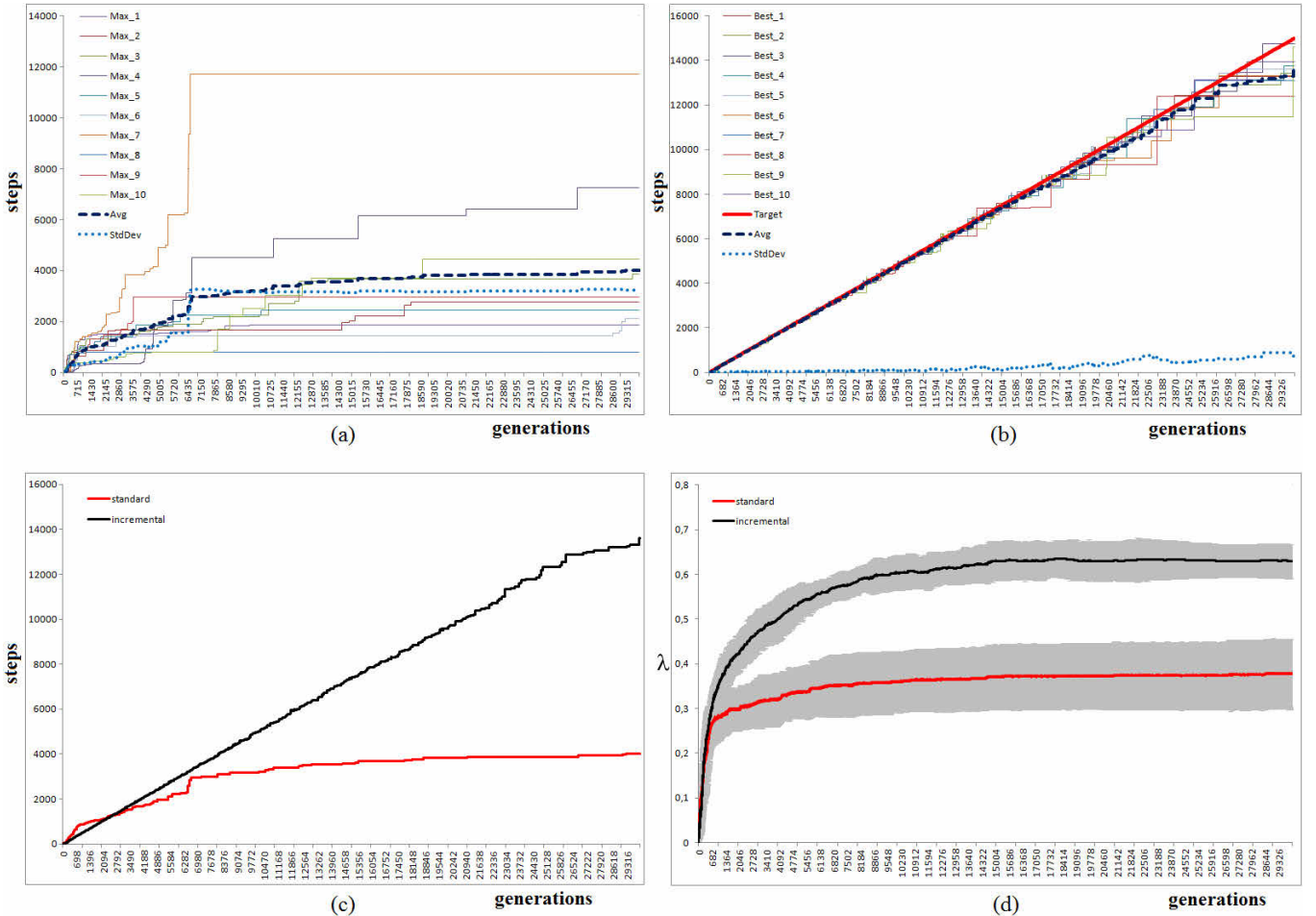
The first set of experiments consists in comparing the behavior of standard GA and incremental GA, starting from dead genomes. In both cases, the GAs run for 30000 generations. The standard approach targets the maximum attractor length for all the 30000 generations whether the incremental approach increments the target attractor length by 10 development steps every 20 generations. It is noticeable that there are clear advantages with an incremental approach for the evolution of complexity.

In Figure 4 (a) the results of a canonical GA are presented. Here the target was the maximum complexity. It is clear that a standard approach could discover in some early generations

good candidates but often the algorithm gets trapped in some unfruitful regions of the search space and for several generations there are no improvements. All the single samples are represented by the thin lines. The dashed line is the average over all the tests and after 30000 generations the attractor length has a value around 4000 development steps. Here the deviation is quite big, since in some cases the maximum length is far from the average. For example, in one case it is close to 12000 development steps while in many other cases is lower than 2000. In Figure 4 (b) the results for the incremental approach are plotted. The straight line represents the target complexity value for each generation, measured as number of development steps inside the attractor. The thin lines here follow quite accurately the target line. The dashed line represents again the average. It is possible to see that average and target are overlapping for the first 15000 generations, whether in the last 15000 generations the average is slightly lower than the target. Overall, the average attractor length after 30000 generations is around 13000 development steps. Here the deviation, represented by the dotted line, is quite small. This means that the incremental approach is able to minimize the distance from the target in each generation. Figure 4 (c) is a comparison of the two different strategies.

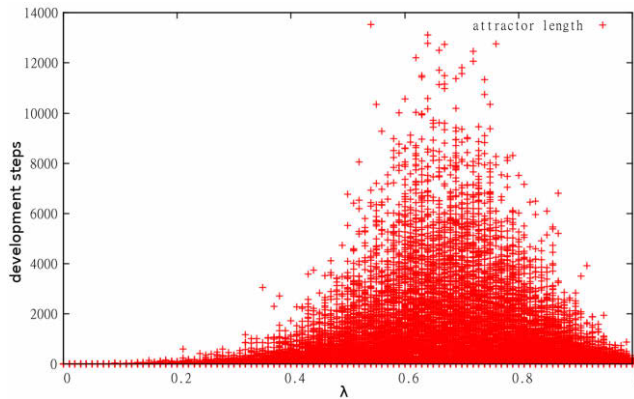
It is evident that the incremental approach overcomes the standard approach since the first 2500 generations. After that, the canonical GA struggles to find good solutions on average and has difficulties to evolve and jump up in complexity. On the other hand, the incremental approach shows very promising results even if in the last generations there is a small degradation of the performances. Overall, the difference between the averages is significant ( $p < .0001$ , Student's t-test).

Finally, Figure 4 (d) represents a comparison of the measured Lambda parameter in each generation. The standard GA evolves to genotypes with a maximum parameter value of 0.4. The incremental GA discovers genotypes with Lambda between 0.6 and 0.7. This means that those genotypes are in a completely different behavioral regime. Earlier work from Langton [16] identified a critical value of Lambda where the behavioral regime of the system encounters a phase transition between ordered and chaotic dynamics. In such area of the search space it is more likely to find the primitive functions to support computation: transmission, storage and modification of information. Further support to this hypothesis is given by previous work on the investigation of probable relationship between attractor length and Lambda [10]. In the experiments herein,  $\lambda$  is used as a measurement and its increase is not a



**Fig. 4:** Results for Experiment 1, developmental tables initialized with “dead genomes”. Avg. and std. dev. over 10 runs. (a) standard GA approach: generations vs. attractor length (thin lines represent single runs); (b) incremental GA approach: generations vs. attractor length (thin lines represent single runs); (c) comparison of averages: standard GA approach vs. incremental GA approach; (d) comparison of Lambda parameter:  $\lambda$  for standard GA approach vs.  $\lambda$  for incremental GA approach.

goal in itself. Figure 5 shows a plot over the  $\lambda$  space where genotypes were generated according to a specific parameter value and the resulting attractor length was measured.



**Fig. 5:** attractor length as function of  $\lambda$ . Results for 4x4 organisms and 1000 tests for each  $\lambda$  value. Adapted from [10].

In those experiments, the cellular automata configuration was the same as in the experiments herein: 2-dimensional grid with neighborhood of size 5 and 3 possible cell types. With such configuration, the most heterogeneous genotypes are generated when  $\lambda$  is 0.66. In fact, in the scattered plot, it is more likely to find long attractors in that area of the solution space. On the other hand, when  $\lambda$  is around 0.4, the behavioral regime is in an intermediate region where organisms show ordered dynamics. As such, it may be more challenging to evolve towards longer attractor lengths. Relating this results with those in Figure 4 (d), it is possible to conclude that the standard GA gets trapped in an area of the search space where the sought behavior (maximum complexity) is less likely to be found. Moreover, it may be difficult for the GA to escape from such region of the search space. The incremental approach is able to evolve genotypes with parameter value around 0.65, which may be beneficial to find longer attractors. Even if not so good solutions are found in that area of the search space, it may be still more probable for the GA to be able to discover better solutions, since the sought behavior is more likely to appear.

## Extended Evolutionary Time

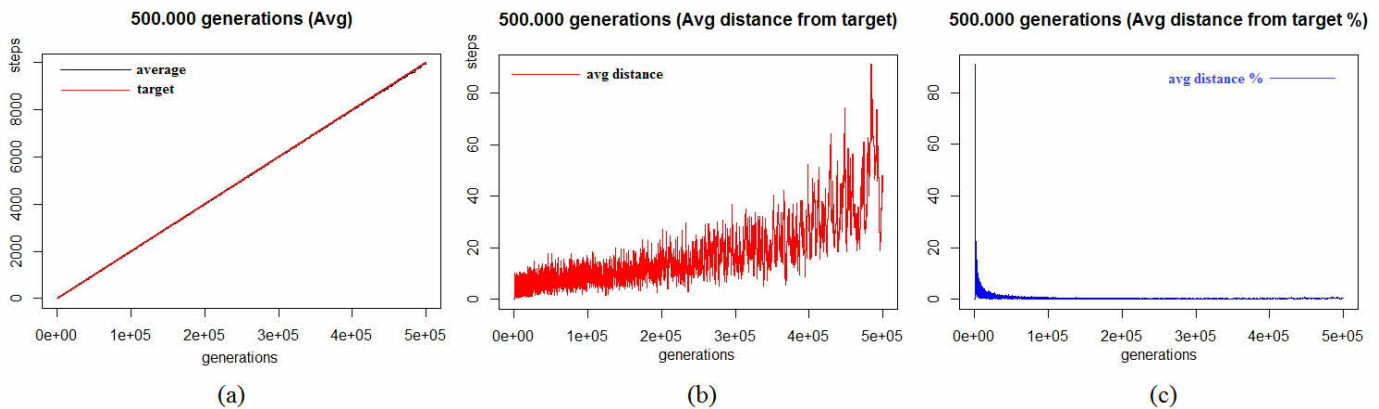
Subsequently, the incremental GA is executed again for 500.000 generations and the target is incremented by 10 development steps every 500 generations. This is done to check the behavior of the GA when the algorithm has more generations to evolve and adapt the population towards the new complexity value.

In Figure 6 (a) the target line and the actual line (average) are completely overlapping. This means that, given enough time to the population to evolve to the sought complexity level, it is possible to keep increase the complexity with minimum deviation from the target. In each generation interval, the genetic algorithm is able to discover favorable genes and use it as a starting point for the next intervals. Evolution is based on already present capabilities, developed incrementally.

Figure 6 (b) and Figure 6 (c) respectively represent the average distance from the target and the percent average distance from the target. It is possible to observe that such span predictably increases along the 500000 generations. Even that, the average distance from the target level suddenly decreases below 1% since the first generations. In conclusion, it may be possible to tune the generation intervals in a way that evolution has enough time to evolve the whole population and prepare it to the following complexity improvements.

## Experiment 2: Randomized Genomes

In the second set of experiments, the behavior of standard GA and incremental GA is tested again for 50000 generations, starting from random genomes, i.e. genomes initialized with a uniform random distribution among the three cell types. By doing that, the standard GA proceeds to select the individual in the population with longest attractor length, targeting maximum complexity. As such, in Figure 7 (a) the average attractor length does not start from the origin. During the first few generations there are several jumps in complexity but after this fruitful stage the average line stabilizes and tends to become flat. Figure 7 (b) summarizes the results for the incremental approach. In this case, since complexity is evolved gradually, individuals with long attractor lengths are left aside in favor of individuals that are closer to the sought initial behavior, i.e. point attractor.



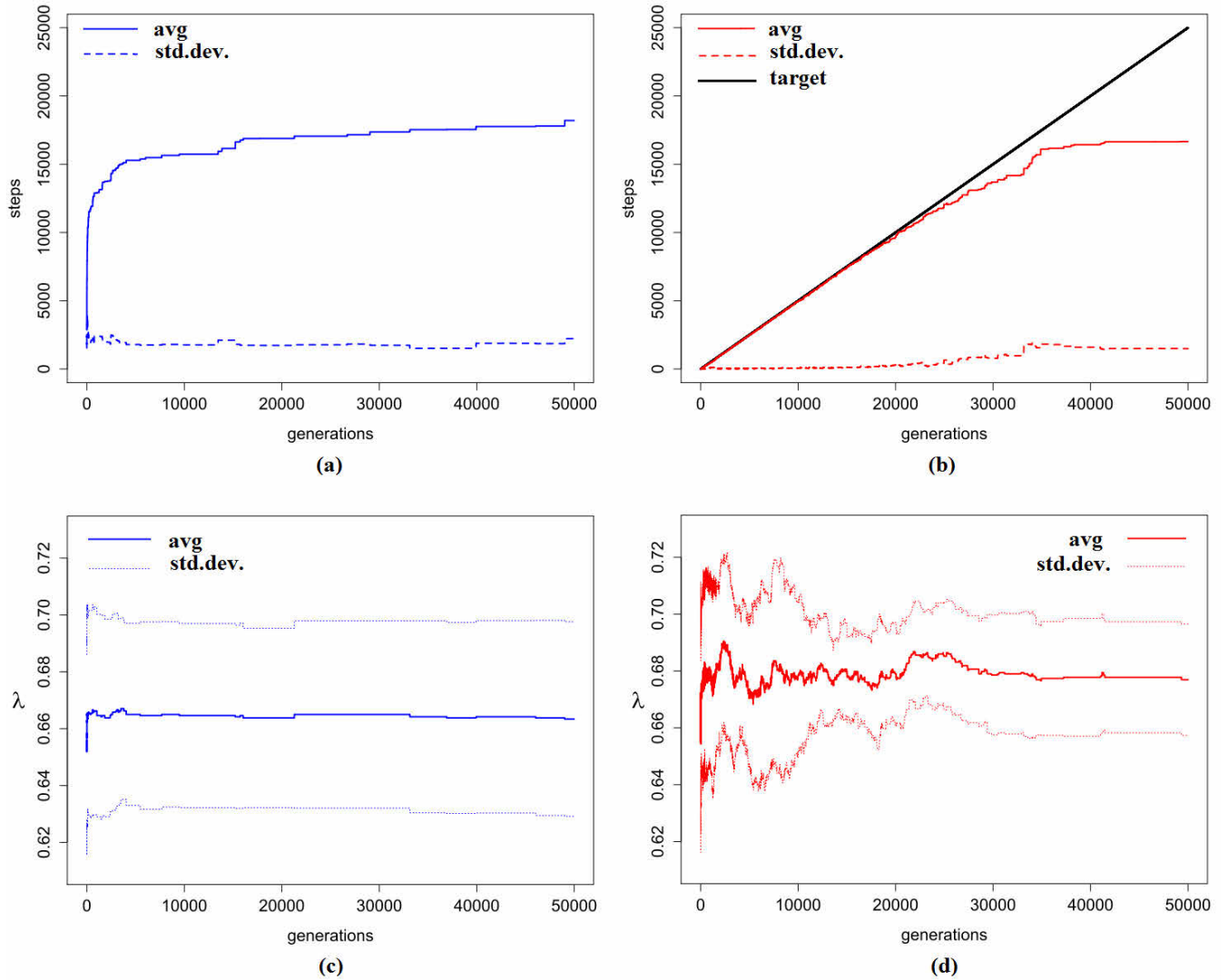
**Fig. 6:** Incremental GA approach with extended evolutionary time (500000 generations), developmental tables initialized with “dead genomes”, average over 6 runs. (a) Target attractor length and actual attractor length (overlapping); (b) Average distance from target in development steps; (c) Average distance from target %



Implicitly, in the beginning the algorithm is forcing organisms to exhibit low complexity to survive, since only in the subsequent generations the environment will become more demanding and will require evolving in complexity. During the first 25000 generations the target line and the actual line are very close. In the last 25000 generations the reached complexity level is very similar to the GA with standard fitness, both in terms of average attractor length and standard deviation. The difference between the averages is not significant ( $p=.0850$ , Student's t-test).

Figure 7 (c) and 7 (d) show the average  $\lambda$  along the generations. Since genotypes were initialized randomly, it is more likely that the developmental tables are the most heterogeneous [16].

As a result,  $\lambda$  is positioned already in an area of the solution space where highly complex individual are likely to appear. For the standard GA, the average  $\lambda$  is smoother and does not show high activity of the GA. Once the algorithm finds good solutions, it is more probable that will hardly improve with better solutions. This could be a drawback for evolvability, especially if one would like to evolve intermediate complexity levels. On the other hand, for the incremental approach,  $\lambda$  fluctuates more within the behavioral region, exhibiting high activity of the GA that continues to explore the solution space. Moreover, the incremental strategy would fit better if the system would need to reach intermediate complexity levels.



**Fig. 7:** Results for Experiment 2, developmental tables initialized with random genomes. Average and standard deviation over 10 runs. (a) standard GA approach: generations vs. attractor length (avg. and dev.); (b) incremental GA approach: generations vs. attractor length (avg. and dev.); (c) Lambda parameter for standard GA approach (avg. and dev.); (d) Lambda parameter for incremental GA approach (avg. and dev.).



## Conclusion

The presented experiments investigated the emergence of complexity in cellular machines, using two different evolutionary strategies: a standard approach, where the target was the maximum complexity and an incremental approach, where complexity was gradually evolved.

We showed that the incremental approach has clear advantages when evolution targets an increase of complexity, especially when the population's genome parameter is towards uniform, i.e. intermediate complexity levels in equilibrium with the environmental pressure. Such knowledge is important at the design stage of EvoDevo systems, where developmental actions are not manually programmed but discovered through evolutionary processes.

We also proposed the usage of Lambda genome parameter to detect the behavioral regime. This may be useful to indicate if the evolving genomes are likely to be able to achieve more complex behaviors, giving information on the evolvability of the system. Such ability of adaptive evolution is necessary for a system to be able to evolve complexity.

Moreover, when it comes to adaptivity, the results herein show that genomes with a given parameter value will most likely mutate to genomes with similar developmental behavior, as long as the mutation results in an offspring with similar parameter value. Our current work is focused on the usage of genome parameters to guide evolution towards favorable areas of the solution space. Furthermore, genome parameters may help to keep the population closer to the desired developmental behavior and supervising its genetic distance. This is in tune with at least two points of the current challenges in the field of artificial life [4]: "explain how rules and symbols are generated from physical dynamics" and "develop a theory of information processing, information flow and information generation for evolving systems".

As a future work, it may be possible to compare the robustness of solutions evolved incrementally versus solutions evolved with a standard approach. In particular, how fragile they are to external perturbation, both at genotype level, i.e. mutations in the rule table, and at phenotype level, i.e. perturbation of the system state during development.

## References

- [1] F. Gomez and R. Miikkulainen. Incremental evolution of complex general behavior. *Adaptive Behavior*, 5: pages 317-342 (1997).
- [2] M. Schuster and C. Harman. Incremental evolution of complex light switching behavior. *CS 81 Adaptive Robotics*. Swarthmore College (2006).
- [3] K. O. Stanley and R. Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of artificial intelligence research*, 21: pages 63-100 (2004).
- [4] M. Bedau, J. McCaskill, N. Packard, S. Rasmussen, C. Adami, D. Green, T. Ikegami, K. Kaneko and T. Ray. Open problems in artificial life. *Artificial Life*, 6: pages 363-373 (2000).
- [5] T. Kowaliw. Measures of complexity for artificial embryogeny. *GECCO 2008*, pages 843-850. ACM (2008)
- [6] P. K. Lehre and P. C. Haddow. Developmental mappings and phenotypic complexity. In *Congress on Evolutionary Computation, CEC2003*, pages 62-68. IEEE (2003).
- [7] R. Doursat, H. Sayama and O. Michel. *Morphogenetic engineering: toward programmable complex systems*. Understanding Complex Systems Series, Springer-Verlag (2012).
- [8] R. Doursat, C. Sánchez, R. Dordea, D. Fourquet and T. Kowaliw. Embryomorph engineering: emergent innovation through evolutionary development. In *Morphogenetic Engineering: Toward Programmable Complex Systems*, R. Doursat, H. Sayama and O. Michel, pages 275-311. "Understanding Complex Systems" Series, Springer-Verlag (2012).
- [9] S. Harding and W. Banzhaf. *Organic Computing*, chapter Artificial Development, pages 201-220. Springer Verlag (2008).
- [10] G. Tufte and S. Nichele. On the correlations between developmental diversity and genomic composition. *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011*, pages 1507-1514. ACM (2011).
- [11] S. Nichele and G. Tufte. Genome parameters as information to forecast emergent developmental behaviors. *11th International Conference on Unconventional Computation and Natural Computation, UCN 2012*, pages 186-197. Springer (2012).
- [12] D. Floreano and S. Nolfi. God save the red queen! Competition in co-evolutionary robotics. *Evolutionary computation*, 5 (1997).
- [13] S. Wolfram. Universality and complexity in CA. *Physica D*, 10 (1-2): pages 1-35 (1984).
- [14] M. Kirschner and J. Gerhart. *The plausibility of life: resolving Darwin's dilemma*. Yale University press (2006).
- [15] U. Krohs and O. Kroes. *Functions in biological and artificial worlds*. The Vienna series in theoretical biology. MIT press (2009).
- [16] C. G. Langton. Computation at the edge of chaos: phase transitions and emergent computation. In S. Forrest, editor, *Emergent Computation*, pages 12-37. MIT Press (1991).
- [17] C. G. Langton. Self-reproduction in cellular automata. *Physica D*, 10: pages 135-144 (1984).
- [18] M. J. West-Eberhard. *Developmental plasticity and evolution*. Oxford Univ. Press (2003).
- [19] G. Tufte. Evolution, development and environment toward adaptation through phenotypic plasticity and exploitation of external information. S. Bullock, J. Noble, R. Watson, and M. A. Bedau, editors, *Artificial Life XI*, pages 624-631. MIT Press, Cambridge, MA (2008).
- [20] G. de Oliveira, P. de Oliveira and N. Omar. Definition and application of a five-parameter characterization of one-dimensional cellular automata rule space. *MIT, Artificial Life7*: pages 277-301 (2001).
- [21] W. Li. Phenomenology of nonlocal cellular automata. *Santa Fe Institute. Journal of Statistical Physics*, 68(5-6): pages 829-882 (1992).
- [22] P. M. Binder. A phase diagram for elementary cellular automata. *Complex Systems*, 7, pages 241-247 (1993).
- [23] P. M. Binder. Parametric ordering of complex systems. *Physical Review E*, vol. 49 n. 3, pages 2023-2025 (1994).
- [24] G. de Oliveira, P. de Oliveira and N. Omar. Guidelines for dynamics-based parameterization of one-dimensional cellular automata rule space. *John Wiley & Sons, Inc. Vol. 6, No. 2 Complexity* (2001).
- [25] T. Kowaliw, P. Grogono and N. Kharma. Environment as a spatial constraint on the growth of structural form. *GECCO 2007*, pages 1037-1044, New York, NY, USA (2007).
- [26] J. F. Miller and W. Banzhaf. Evolving the program for a cell: from french flag to boolean circuits. In S. Kumar and P. J. Bentley, editors, *On Growth, Form and Computers*, pages 278-301. Elsevier Limited Oxford UK (2003).
- [27] R. Dawkins. *The blind watchmaker*. Chapter 3: accumulating small changes. Norton & Company Inc. (1986).
- [28] M. Mitchell, P. T. Hraber and J. P. Crutchfield. Revisiting the Edge of Chaos: evolving cellular automata to perform computations. *Complex Systems 7*: pages 89-130 (1993).

# The ALife Zoo: cross-browser, platform-agnostic hosting of Artificial Life simulations

Simon Hickinbotham, Michael Weeks and James Austin

University of York, Computer Science Department, York, UK  
simon.hickinbotham@york.ac.uk

## Abstract

We describe a new approach to sharing software simulations that is of great potential benefit to Artificial Life researchers. youShare is an online collaborative facility that allows users to upload data, and software in the form of services. An attached execution environment allows services to be run over a heterogeneous cluster of compute nodes, where the service infrastructure guarantees that the service will be executed in the correct environment, and provide consistent results. It allows software to be made available as a service regardless of the operating system they run upon. This allows software to be maintained more easily, and to be available to all researchers with internet access. We demonstrate this by making three Artificial Life simulations available over the web: Tierra, Avida and Stringmol. These services form the foundation of an ALife “Zoo”, in which visitors can interact with ALife simulations for research and education. In addition, youShare offers a workflow facility whereby multiple services can be connected to create more complex tasks. We demonstrate the utility of this system in Artificial Life research via a workflow which calculates evolutionary activity for runs of Tierra and Stringmol.

## Introduction

It is common practice for researchers and scientists that develop their own algorithms and programs, to make them available to other researchers as source code and/or binaries. Often these are distributed to the community via personal, community or commercial websites. With a limited range of operating system (OS) configurations available to the originators of the software, it is very common that third-party developers who attempt to compile and execute these applications experience compile-time errors, dependency errors and run-time errors.

One solution is to develop and maintain source code or binaries that run on a range of operating systems. There are several problems with this approach:

- it is time consuming for the developer, who is often only experienced in writing software for personal use whilst pursuing their own research;
- considerable expertise is required to compile binaries or

install software, particularly when it has not been written by an expert in writing production-quality software;

- the code base becomes increasingly unwieldy, making development by the original authors or the community more difficult.

It should also be noted that if researchers do not make their code available, it can promote the independent development of code bases that may not conform to the original algorithm specification. The lack of availability of source code, or difficulties in compiling code if it is available, have the potential to reduce the impact of the research. The onus is therefore upon the researcher to make software available, which places further burden on a finite resource, namely the researcher’s time.

Researchers in Artificial Life (ALife) will be particularly familiar with these issues, not least due to the interdisciplinary nature of the field. A cross-disciplinary research team will have fewer skilled programmers than a pure computer science project, and fewer domain experts than a pure biological project. There are myriad ALife computer simulations available (for a list, see [http://en.wikipedia.org/wiki/Artificial\\_life#Notable\\_simulators](http://en.wikipedia.org/wiki/Artificial_life#Notable_simulators)), many of which share common themes and research applications. However, research that explores ALife issues using more than one of these simulators is rare.

In this contribution, we demonstrate a system that is capable of overcoming these issues, based upon the YouShare Virtual Laboratory (<https://portal.youshare.ac.uk>)(Austin et al., 2011). YouShare is an online collaborative facility that allows users to upload data, and software in the form of services. An attached execution environment allows services to be run over a heterogeneous cluster of compute nodes, where the service infrastructure guarantees that the service will be executed in the correct environment, and provide consistent results. Of particular interest to the ALife community may be the workflow facility that allows multiple services to be connected together to create more complex scenarios. To demonstrate this we

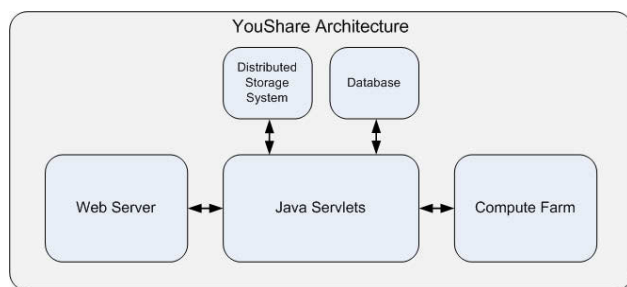


Figure 1: Overview of the YouShare Platform Architecture

have wrapped instances of a small set of representative simulators and made them publicly accessible to all YouShare users. In addition, we have created two auxiliary data analysis services that can be used on comma-delimited data files that are easily produced from the simulators.

### YouShare Platform

YouShare is an internet-based Virtual Laboratory Environment (VLE) that provides a collaborative repository for data, executable software services and workflows. The YouShare platform is based upon work achieved on the CARMEN Neuroscience platform (<http://www.carmen.org.uk>) (Watson et al., 2007) to provide a VLE for neuroscientists. YouShare extends CARMEN to be a generic, cross-domain platform for UK academics and researchers.

The YouShare platform is based upon a standard three-tier web architecture (Eckerson, 1995), as shown in figure 1. The first-tier, also known as the presentation layer, consists of a web portal, providing the user with access to the system via a web browser. The portal is built using the Google Web Toolkit (GWT), which accommodates inconsistencies in the implementations of Javascript between all the web browsers in common use today. The application layer (tier 2) consists of a group of Java Servlets, as defined by Oracle (Oracle, 2013). These provide an API onto the Data Storage Tier, and provide access to the compute servers, which are typically 8-core zeon virtual machines with 16GB RAM. The data storage layer (tier 3) comprises an SQL type database and a storage system. The database contains user accounts, metadata, system states, and the relationships between users and artefacts (files, services, metadata, workflows, etc). The storage system is used to store user's files, service implementations, and workflows.

YouShare is accessed via the portal (<https://portal.youshare.ac.uk>); a login facility provides secure access for users. Data can be uploaded to the YouShare platform, along with fully descriptive metadata (Jessop et al., 2010). Once uploaded, sharing permissions can be applied to both the data and the metadata. Several levels of sharing rights are available: private to the user; shared with specific people or groups of people; or public to

all registered users of YouShare.

Software applications can be deployed to the platform in the form of executable services. These services can be used to analyse private, shared or public data from the data repository. A service consists of the software implementation and a metadata description of the service. The services are executed on a heterogeneous array of compute servers attached to the YouShare platform. Services can be executed from the YouShare portal individually, or combined into a workflow to create more complex processing scenarios. The owners of YouShare services and workflows can set sharing permissions in a similar way to data. To aid discovery of suitable data, services and workflow, an Apache Lucene based search engine is provided to search each of the repositories.

### YouShare Services

A YouShare Service (Weeks et al. (2013)) is a combination of software that can be executed on the YouShare platform, and a metadata document to describe its operation and implementation. The software can be based upon a users proprietary code, or a common software package these can be converted into a service via a wrapping process using the YouShare Service Builder application, then deployed on the YouShare platform. We can currently create services from programs written in Python, Java, R, C/C++, Matlab, Perl, Bash/Bat scripts, or indeed most executables or scripts that can be run via a command line. YouShare Services also support multiple OS platforms, such as Linux and Windows.

The Service metadata document is a key component in the service. It provides the name and description of the service so that user's can select suitable services for their requirements. A description of the input parameters provides an interface on the portal when a service execution request is made. This interface provides a description for each input, and a suitable means to set the inputs, i.e. file browser, text box, or drop-down menu. Platform and environment specific information for the service execution can be set, to ensure that service is deployed onto the correct type of compute server, along with any data that needs to be staged. This ensures correct execution, and completely removes the need for user's to know the implementation details. Upon successful execution of the service, the metadata is used to stage the results back into the data repository and to provide a portal interface to display the output results.

### Creating Services

One of the principles behind YouShare is that users can create their own services. To this end, the service creation process needs to be quick and easy to achieve. We also need to support multiple operating systems and application programming types, and need a common method for passing data into and out of the software. These requirements can be achieved by converting the software into a command line application. Input parameters are passed into the application

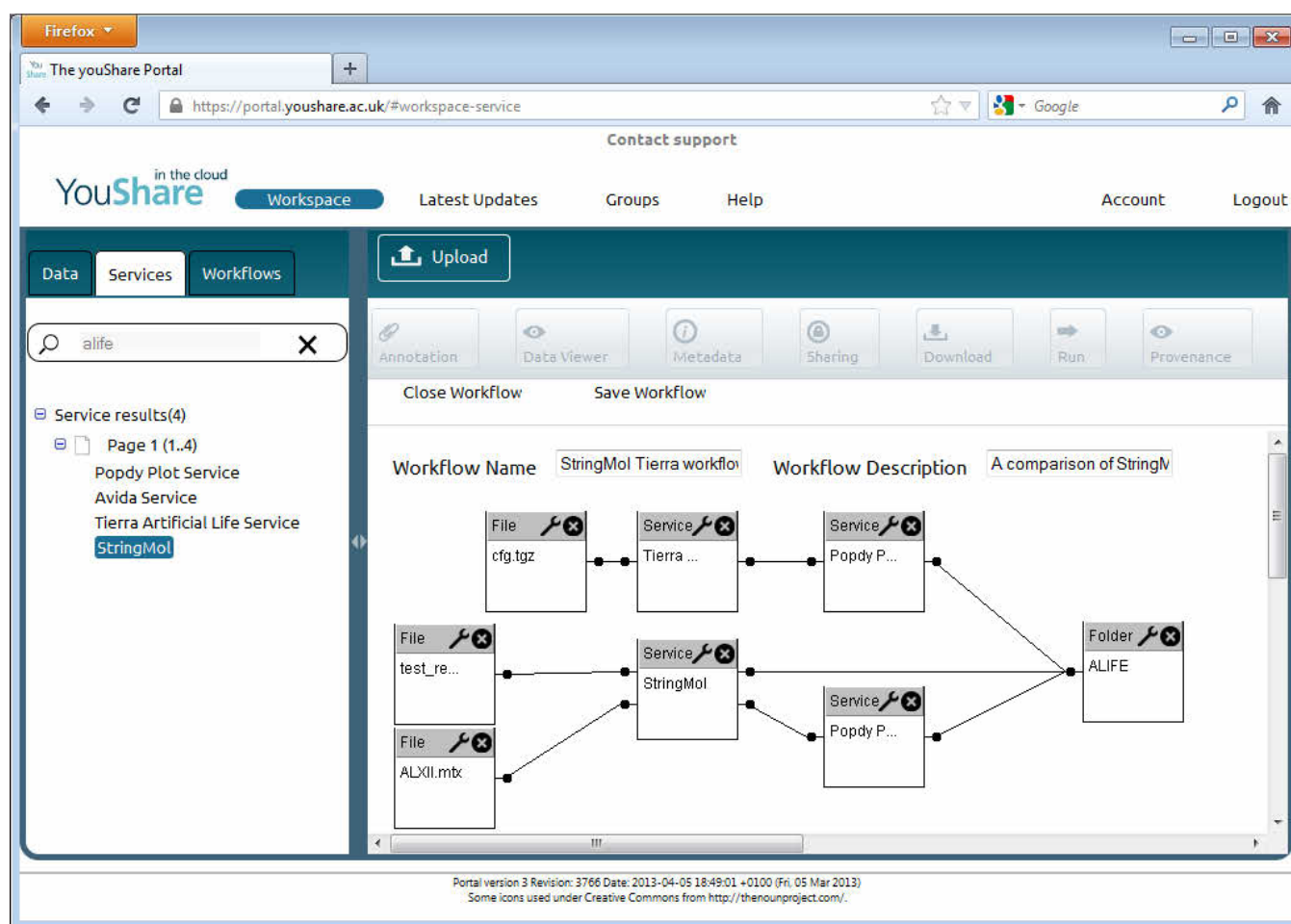


Figure 2: Example YouShare workflow consisting of Stringmol and (a modified) Tierra services being analysed by the Stringmol popdy plotting service. The results from both services are placed in the same output folder

via the command line parameter list. On completion of the application, the service needs to register its outputs in the database. This is achieved by getting the application to print any output values and/or file names to the screen (stdout) in a comma-separated list, surrounded by xml `<output>` tags, which can be achieved either by simple changes to the source code of the application, or by calling the application from a script which prints the output data once the application has finished.

To enable the command-line application to be interfaced with YouShare's Java Servlets (application layer), the application is encapsulated within a small Java class wrapper. A metadata document is also necessary to describe the application/service from a user and system standpoint. We have developed a tool, called the YouShare Service Builder, to automatically generate the wrapping and metadata. The Service Builder is a Java standalone application which uses a wizard-based approach using simple forms. The Service Builder generates a jar (Java archive) file containing the ser-

vice implementation, and an XML file containing the meta-data document. These files are uploaded to an admin panel on the portal in order to deploy the service.

## Running Services

To run a service, the user must first discover it. A search facility allows users to search the service repository, and displays a list of matching services that are accessible to that user depending on the service sharing settings. The user can browse the list of returned services and select one that is suitable. The service description can be viewed, and the service bookmarked for later use, or executed by selecting the "run" button which displays the service execution panel. This panel displays the input parameters to the service as described by the service metadata, and allows the user to select suitable data inputs. Once the inputs are set, the user can select a folder in their data repository for the results to be placed, and the service execution is then initiated. Whilst running the service execution progress is displayed



in the service log, where, upon completion the results are displayed. The results can also be viewed by browsing the specified results folder in the data repository.

## Workflows

Services can also be executed within the YouShare workflow facility that is available on the portal. Workflows allow more complex analyses to be performed, using a collection of suitable services, connected together in serial and/or parallel. The YouShare workflow tool is based on two components; a graphical workflow editor embedded into the portal, and a workflow enactment engine in the application layer servlets. The workflow editor allows users to place and connect input files, services, and output folders on the workflow editor panel. Once created a workflow can be saved and reloaded for later use. Modifying a workflow creates a newer version, though previous versions can be reloaded very simply. A workflow can be shared with other users, or groups, in the same way as files and services, however each component of the workflow must have a suitable sharing permission for a user to use a shared workflow. An example YouShare workflow is shown in figure 2. Once a workflow is ready for execution, it must be saved and the workflow editor closed. Selecting the "run" button, passes the workflow script to the workflow enactment engine. The enactment engine builds a model of the workflow, and orchestrates the service execution, and the flow of data between the service processes. Since a service implementation details are handled via the service API via the service metadata, the workflow can be constructed from a combination of services that each require different platforms and environments.

## The ALife Zoo

Having described the youShare system, we now turn to its utilisation in ALife. Our principle aim is to create an ALife "Zoo" consisting of simulators and associated data that are available to run on a single, free resource, and to foster the development of common methods of analysis of their outputs. The first step towards this goal is to make simulators available as services. In this section, we detail how this has been achieved for Tierra, Avida and Stringmol.

### Tierra as a service

The first paper on Tierra appeared in 1991 (Ray, 1991) - Tierra is close to a quarter of a century old, yet it remains highly influential. The first paper has 931 citations on google scholar. Research on Tierra is still active (Shao and Ray, 2010). Anecdotal, during the panel discussion at ECAL 2011, one of the topics was 'most influential work' - the panel unanimously credited Tierra.

**Building the core service** We downloaded Tierra version 6.02, with the patch from Matthias Rav (<http://tierra.lolwh.at/>), and compiled it on one of our CentOS5 64-bit

Linux compute servers. Rather than modify the C source code to suit our requirements, we created a bash script to wrap the Tierra command line application and provide the XML <output> tags. This ensures that the service runs as the authors of the original software intended.

One of the limitations of our service framework is that a service must have a fixed number of inputs and outputs. This can be a problem where the software that the service is built from can be configured with varying numbers of arguments. Tierra takes a group of configuration files as it's input data, and produces a number of files in an output folder. To solve this, the bash script takes in a single zip or tgz archive of input configuration files, and un-archives them to the working folder on the execution server. Similarly, the output folder is archived by the bash script to produce a single output tgz file.

When Tierra fails it does so in an interactive way, asking the user for input. As this will be executing on a remote server, Tierra would not get a user response and so the service would hang, eventually timing out. Ideally, we want the Tierra service to terminate on error, so we modified the Tierra source code by commenting out the interactive code. To provide additional robustness, the bash script performs some degree of pre-checking to see if the configuration files are complete. The Tierra command line and it's calling bash script were then wrapped using the Service Builder tool, and the resulting service was deployed on YouShare with public sharing access.

### Avida as a service

Unlike Tierra, Avida (Adami and Brown, 1994; Fortuna et al., 2013) assigns every digital organism its own protected region of memory, and executes it with a separate virtual CPU. By default, other digital organisms cannot access this memory space, neither for reading nor for writing, and cannot execute code that is not in their own memory space. A second major difference is that the virtual CPUs of different organisms can run at different speeds, such that one organism executes, for example, twice as many instructions in the same time interval as another organism. The speed at which a virtual CPU runs is determined by a number of factors, but most importantly, by the tasks that the organism performs: logical computations that the organisms can carry out to reap extra CPU speed as bonus.

**Building the Avida service** Avida is very similar in operation to Tierra, in that there are multiple input configuration files and multiple output files. The Avida C++ source code was compiled for our CentOS5 Linux environment, and a similar bash script was used to encapsulate the Avida command line executable. This script is used to pass the configuration files into the service as a single input archive file, and the output files as a single archive file. The bash script also performed some degree of checking for correctness on



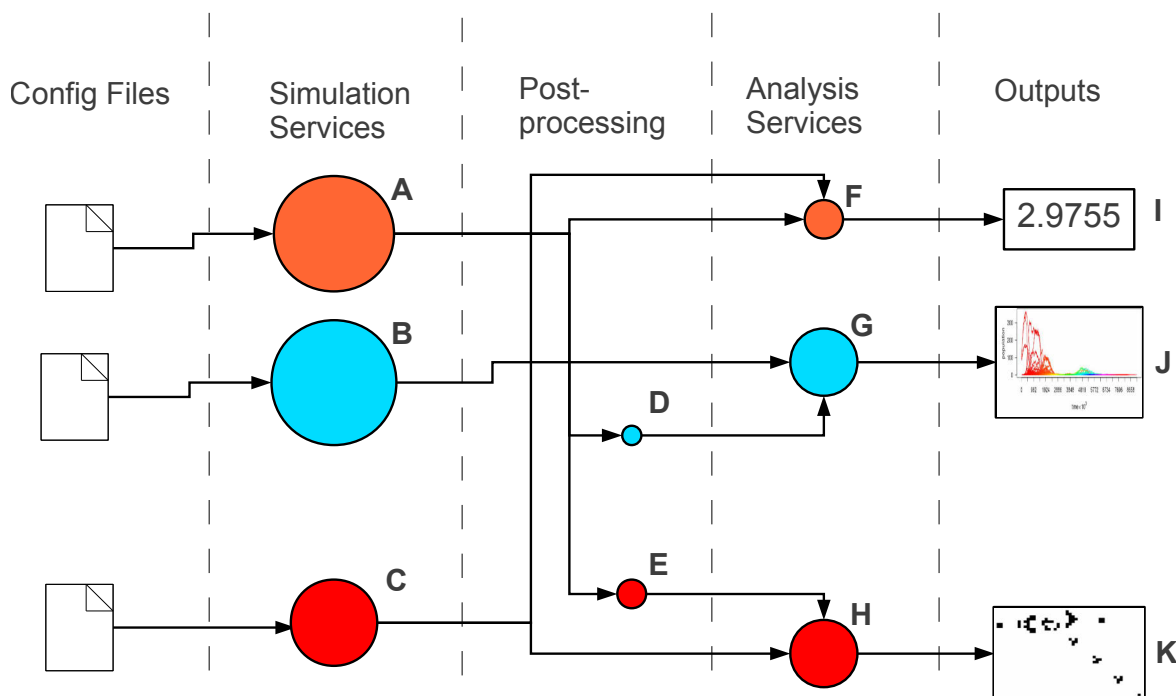


Figure 3: Hypothetical illustration of the set of possible workflows using services developed by three research groups, shown in Orange, Blue and Red. Each service is represented by a circle. The size of the circle indicates the effort involved to develop the original software (effort to wrap the software into a service is negligible). An individual workflow is constructed by connecting services together following arrows from left to right. Services developed by different teams can be linked together provided they are compatible. For example, service A can only be linked to service G if it is passed through post processing service D.

the input configuration files. We wrapped the bash script and Avida version 2.12.4 (Anon., 2012a) into a service and deployed them on YouShare with public sharing rights.

### Stringmol as a service

Stringmol is a more recent simulator which uses a much simpler model of the individual (Hickinbotham et al., 2010, 2012b). In Stringmol, there is no virtual CPU and all operations are carried out on the sequence of the individual, rather than storing values in a stack. Despite the simplicity of the implementation, Stringmol is capable of generating suprisingly innovative programs.

**Service implementation** Stingmol has not had the extensive software development that Tierra and Avida have undergone and so was much more straightforward to implement. Stringmol is written in the C++ programming language, and we compiled version 0.2.1 for our CentOS5 Linux servers, and wrapped it as a service. Two input files must be specified, plus a numerical selection of the type of simulation that is to be executed (bi-molecular interaction, simulation of a single container of molecules, or simulation of a population

of containers of molecules). However, the number of output files that are created depends on the configuration of the simulation. A script was used to collect these output files into a single zipped file that forms the output of the service.

### Analysis services

Although the three simulators described above run as stand-alone programs, each of them have auxiliary programs that are used to analyse the outputs of the simulation in order to carry out research on their behaviour. The analysis program need not be written in the same programming language as the simulator, nor need it run on the same operating system.

To illustrate how these analyses can be coupled to simulators in a workflow, we have deployed two analysis services, written in the R language. These services were originally written to analyse the outputs of the Stringmol simulator, but we have adapted Tierra to produce output that can also be analysed by these services. This approach allows analysis tools to be developed which can be used to evaluate and compare the different simulators. The first analysis service produces graphs of population dynamics of different species

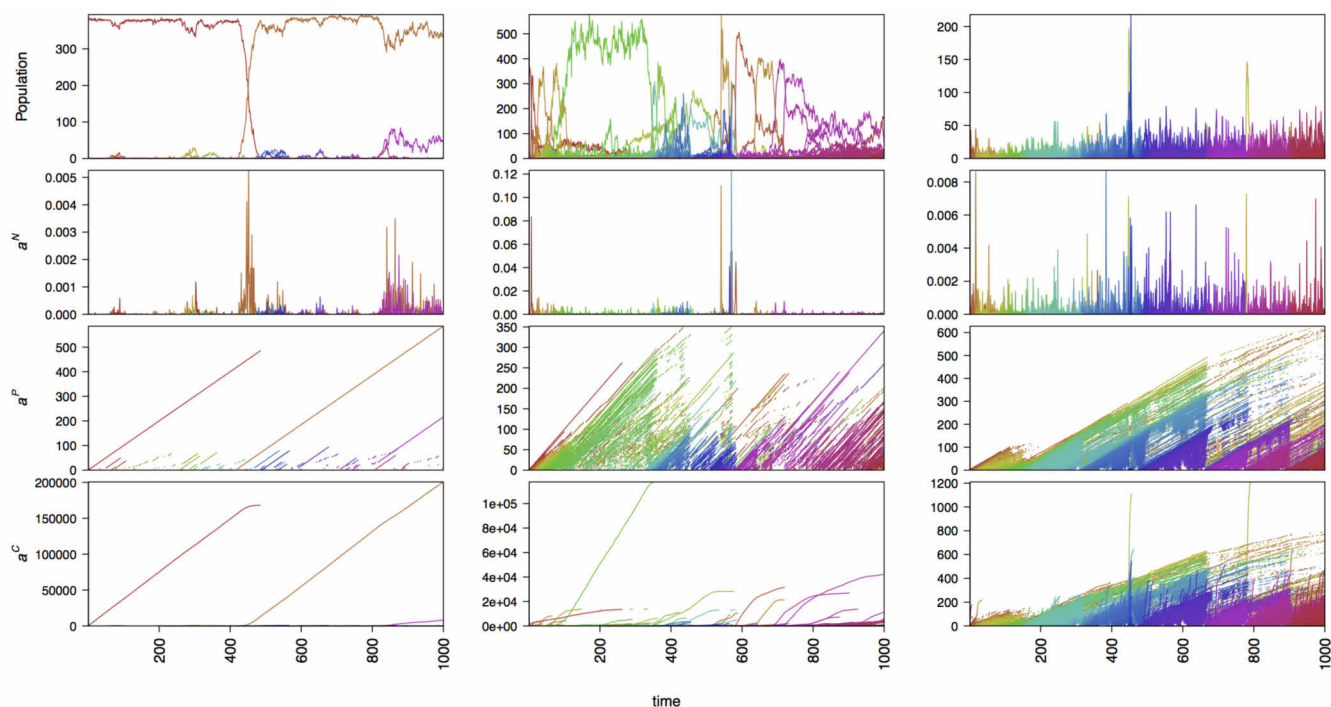


Figure 4: A plot of population dynamics (top) and non-neutral, population- and count-based measures of evolutionary activity, as executed in the workflow in figure 2. See Droop and Hickinbotham (2012) for further details

in the simulator, and the second produces a measure of evolutionary activity (Droop and Hickinbotham, 2012).

### ALife Workflows

Although the provision of software services is a useful aim in itself, the real power of the YouShare system arises when services can be connected together to form workflows. This allows research groups to collaborate on developing common research themes with relatively little effort. Figure 3 illustrates a hypothetical example of a set of workflows that use services originating from three different research teams, shown in different colours in the figure. Each team has developed an ALife simulation and an analysis tool, that have been deployed on the portal. In this example, the analyses take three forms: Numeric data (I), Graphs (J) and animations (K), demonstrating that workflows offer a powerful and efficient way of re-using code, providing a common analysis base between research groups.

#### Example - Analysis of StringMol and Tierra data

As an illustration of the workflow capabilities, we created a scenario comparing evolutionary activity in StringMol and Tierra (see figure 2). The resulting workflow automates the experiment published in Droop and Hickinbotham (2012), in which population dynamics and three measures of evolutionary activity we calculated for configurations of Tierra and StringMol, configured with a range of mutation rates.

To achieve this, we created a modified Tierra service so that it created output in StringMol format. The "popdy plotting service" was then created, based on existing R code that produced visualisation of the StringMol output data. The workflow loads data from a publicly shared location, pushes the data into both StringMol and Tierra, the outputs of which are separately analysed and written to a directory in the current user space.

Upon completion of the workflow execution, the output folder contains two PDF files from the Tierra and StringMol analyses respectively. See figure 4 for the analysis of Tierra data. The top row shows the population dynamics of different component types in Tierra during the run. The second row shows the non-neutral evolutionary activity, and the third and fourth rows show population- and count- based evolutionary activities of Bedau et al. (1997).

The output folder also contains an optional log file that was generated by the StringMol service. This workflow has been shared publicly for all users of YouShare.

### Further Work

For the current contribution, we have implemented the ALife Zoo in the generic YouShare system. This allows a test of concept to be created quickly, and range of other generic services that have been implemented for other research domains to be available. Like many web applications, YouShare undergoes constant improvements as the user and

code base expands. Whilst the ALife Zoo could be hosted by the “vanilla plain” YouShare platform on an ongoing basis, we recognise that there may be special demands for specific features from the ALife research community. It is possible to branch the first tier of the service to deliver a bespoke user experience. We have successfully done this for several projects (e.g. Hickinbotham et al. (2012a)). Below we discuss potential improvements to the core YouShare system that could be implemented in future.

### Visualisation

Where simulations produce large amounts of output data, it is desirable to create a visualisation of the data before deciding to download the data and proceeding to more detailed analysis. This problem has been partially addressed in the CMAC project, where interactive visualisation of the service output is made available on-line using the Raphael Javascript library (Anon., 2012b). The challenge in ALife Zoo is to make these visualisations as informative as possible, whilst keeping them sufficiently generic to be applicable to a range of simulations.

### Linking to Publications

YouShare was developed to function as a Virtual Laboratory, and since publications are a permanent record of the work in Laboratories, it makes sense to link them to their data, services and workflows in YouShare. A facility to make such links is currently under development.

### Linking to External HPC

Although the compute resource in YouShare is considerable, it is not infinite. Certain services may require high-memory or many-core processing to be delivered in a timely manner. YouShare could be extended to carry out the interfacing and staging of jobs on HPC clusters with relative ease, and so combine the advantages of HPC with the advantages of a browser-based user interface. A key issue here is handling big data, but ALife simulations tend to be relatively small in terms of their data footprint.

### Linking to External Tier 1 Server

Finally, it is possible that particular specimens in the ALife Zoo may require special handling in their presentation to the public. In other words, the GWT-based youShare portal may not be sufficiently flexible to demonstrate particular simulations. For example, it would be extremely difficult to implement Avida-Ed, the educational version of Avida, in YouShare, due to the highly graphical and interactive nature of the interface. However, it is possible to develop other browser based front ends (for example, a LifeRay portal) that could access the YouShare back end whilst delivering a highly specialised user interface.

### Exposing services to other web applications

Since the services in youShare are RESTful, it is straightforward to make them available to other web applications, rather than using the generic portal. We are in the process of developing an API to make these services available in this way. The main issue to be addressed is the maintenance and handling of user access rights to youShare.

### Conclusion

YouShare provides a method for collaborative sharing of data and software. Software is deployed as “ready to go” services that are guaranteed to be deployed on the correct platforms of a heterogeneous compute facility. However, service implementation details are hidden from the user, allowing them to concentrate on service functionality alone. Mixed-platform services can be combined in a workflow environment to create more powerful tools. The YouShare compute facility can be extended to the Cloud, and with Virtual Machine (VM) technology we provide a sustainable software model. YouShare also encourages cross-domain research; there are currently 83 services on YouShare, including the new ALife services we describe here. These are a mix of Windows, Scientific Linux 4, and CentOS Linux 5 services, and cover domains such as neuroscience, text mining, image/3D processing, and neural networks, as well as generic services.

Two of the ALife services and workflows we report in this contribution have been highly influential in the field over the past two decades. With relatively little effort, and thanks to their open-source licenses, we have been able to make them available as web services for the first time. We are also able to show how analysis services developed more recently can be used to analyse the outputs of Tierra. This demonstrates that platforms such as YouShare are able to create flows of configuration, simulation and analysis between ALife researchers in a novel and efficient manner.

### Acknowledgements

The youShare project funding is provided by the UK HEFCE University Modernisation Fund. CARMEN was developed under funding provided by the UK EPSRC on contract number EP/E002331/1, and currently supported by the UK BBSRC under contract number BB/1000984/1. CMAC project funding was funded by the UK Technology Strategy Board.

### References

- Adami, C. and Brown, C. (1994). Evolutionary learning in the 2d artificial life systems avida. In Brooks, R. and Maes, P., editors, *ALife IV*, pages 377–381. MIT Press, Cambridge, MA.
- Anon. (2012a). Avida software: <http://sourceforge.net/projects/avida/files/avida-stable/2.12.4/> - retrieved 30th june 2013.

- Anon. (2012b). Raphaelgw: <http://code.google.com/p/raphaelgw/> - retrieved 3rd may 2013.
- Austin, J., Fletcher, M., Jackson, T., Jessop, M., Turner, A., and Weeks, M. (2011). Youshare, an online collaboration research environment for sharing data and services. In *UK e-Science All Hands Meeting 26th-29th September 2011, York, United Kingdom*.
- Bedau, M., Snyder, E., Brown, T., and Packard, N. (1997). A comparison of evolutionary activity in artificial evolving systems and in the biosphere. In *ALife IV*, pages 125–134. MIT Press.
- Droop, A. and Hickinbotham, S. (2012). A quantitative measure of non-neutral evolutionary activity for systems that exhibit intrinsic fitness. In *ALife XIII*. MIT Press.
- Eckerson, W. (1995). Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications. In *Open Information Systems 10*.
- Fortuna, M., Zaman, L., Wagner, A., and Ofria, C. (2013). Evolving digital ecological networks. *PLoS Comput Biol*, 9(3).
- Hickinbotham, S., Austin, J., and McAvoy, J. (2012a). Interactive graphics on large datasets drives remote condition monitoring on a cloud. *J. Phys.: Conf. Ser.*, 364.
- Hickinbotham, S., Clark, E., Stepney, S., Clarke, T., Nellis, A., Pay, M., and Young, P. (2010). Diversity from a monoculture: Effects of mutation-on-copy in a string-based artificial chemistry. In *ALife XII*. MIT Press.
- Hickinbotham, S., Clark, E., Stepney, S., Clarke, T., Nellis, A., Pay, M., and Young, P. (2012b). Specification of the string-mol chemical programming language version 0.2. Technical Report YCS-2010-458, Univ. of York.
- Jessop, M., Weeks, M., and Austin, J. (2010). CARMEN: a practical approach to metadata management. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 368(1926):4147–4159.
- Oracle (2013). Java servlet technology overview : <http://www.oracle.com/technetwork/java/javasee/servlet/index.html> - retrieved 3rd may 2013. Oracle.
- Ray, T. (1991). An approach to the synthesis of life. In *ALife II*, pages 371–408.
- Shao, J. and Ray, T. (2010). Maintenance of species diversity by predation in the tierra system. In *ALife XII*, pages 533–540. MIT Press, Cambridge, MA.
- Watson, P., Jackson, T., Pitsilis, G., Gibson, F., Austin, J., Fletcher, M., Liang, B., and Lord, P. (2007). The CARMEN Neuroscience Server. In *Proceedings of the UK e-Science All hands Meeting*, pages 135–141.
- Weeks, M., Jessop, M., Fletcher, M., Hodge, V., Jackson, T., and Austin, J. (2013). The carmen software as a service infrastructure. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1983).

# A Temporal Pattern Predictor for Virtual Characters

Micah Rosenkind<sup>1</sup>, Prof. David Arnold<sup>1</sup>, Dr. Graham Winstanley<sup>1</sup>

<sup>1</sup>University of Brighton, UK  
m.m.rosenkind@brighton.ac.uk

## Abstract

This paper discusses a prototype of a temporal pattern predictor, which was built on specifications derived from the descriptions of the “Ergotrix” temporal memory network in Valentino Braitenberg’s “Vehicles” (Braitenberg, 1984). The prototype was developed as a component for a control architecture for virtual characters.

## Introduction

In Valentino Braitenberg’s “Vehicles - Experiments in Synthetic Psychology” (Braitenberg, 1984), the author describes a temporal pattern memory component that enables the synthetic creatures to base their behaviour on past experiences: Rather than just reacting to the currently perceived sensory stimuli, these agents first form predictions based on the statistical probability of previously perceived patterns re-occurring. These predictions then form the basis for their actions. In other words, the creature reacts to a pattern of expected stimuli, instead of waiting for the stimulus to actually occur.

Braitenberg’s temporal pattern memory component is implemented in the form of a connectionist network of nodes. In such a network, each node may represent the presence of a sensory stimulus, a pixel on a screen or a sensory ‘value’ of another kind (e.g. the distance value of an infra-red sensor). The individual nodes can potentially form connections (which are often referred to as associations) to any other nodes in the network. The strength of the connection is determined by the connection weight.

In many classical neural network models, such as feed-forward or back propagation networks, (e.g. McCulloch Pitts 1943, Bryson and Ho 1969,), the connection weights are set using sophisticated learning algorithms. However, many of these separate the process of setting weights (learning or training phase), from actually using the network (execution phase). The model presented in this paper, uses a learning model that can form new associations while it is being used. This learning rule is based on Hebb’s “what fires together, wires together” postulate (non-verbatim from Hebb, 1949), which itself tried to summarize the general behaviour of connecting neurons in biological brains.

Regardless of the learning rule used, traditional implementations of connectionist networks will see that a signal passes from active nodes to all their connected peers.

The amount of activity transmitted is determined by the weight of the connection.

In addition to the properties that define a Hebbian learning-based neural network, Braitenberg specifies two further key characteristics, which our network model intends to address:

1. The memory network associates only elements that are active in succession, within a brief delay and not those, which are active simultaneously. This differentiates it from a basic associative connectionist network.
2. Memorized patterns can be reproduced at an arbitrary speed. If they are reproduced at a more rapid pace than they are likely to occur as sensed via the sensory system, the network acts as a predictor.

These two functional requirements were at the core of a series of our incremental prototypical experiments.

## Methods

### A Fixed-Delay Network

After initial investigation into delay-line networks, which used timers to record variable activation delays between a set of elements (see Figure 1), we employed a network with fixed delays between the activation of elements. The previous models had focused on implementing the ability of modelling a variety of time delays between device pairs in order to learn temporal patterns in threshold device populations. The model discussed in this paper moves away from this variable delay paradigm towards a notion inspired by the way film cameras record the passage of time. Film cameras are usually set to operate at a fixed frame-rate; say 25 frames per second (fps). When a slowly moving object passes by the camera lens it leaves a contiguous trace of strongly activated pixels on the film (or CCD chip for that matter). On the other hand, if a fast moving object passes by the camera lens, it would only leave trace signals on a few, further apart areas of the film/CCD chip. Thus the speed of the object is represented by two distinct patterns on the film, even though the delay between the activation of the neighbouring contiguous pixels was the same in both cases. The time delay between the contiguous and distant impressions was 1/25 of a second.

This model uses the idea that speed is not a matter of delaying information transmission between devices. At one time-step the delay between one node activating another is always



constant. Rather, time is a property of an observed object and the pattern that its observation leaves in the connections of a perceiving network. Figure 1 illustrates the fixed delay network. The activation timing between each pair of devices in the network is fixed. The timings are not stored in the connections, but in the patterns that are left when sensed by a group of connected devices.

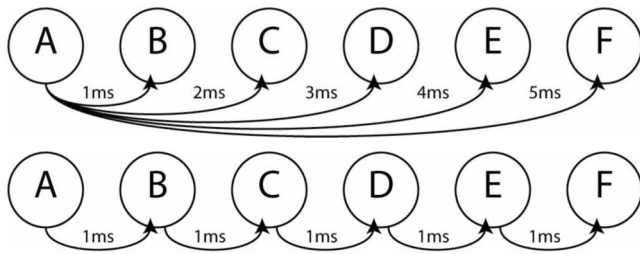


Figure 1 Illustrates variable delay and fixed delay paradigms

Applying the above example to a network of connected nodes ABCDEF, that are aligned in the visual field of a camera from left to right and an object passes by them at slow speed, then all devices A-B-C-D-E-F will be activated in sequence. If the object passes by at high speed, some visual receptors might actually fully activate- in this case only triggering the sequence A-C-E for example.

Thus, even though the activation speed between the threshold devices in the visual receptor is constant, they can capture different timings/speeds within varying received activation patterns.

Comparing this theoretical model to the subjective experience of observing moving objects, an object moving past a photo-cell faster would allow less light to pass from it into any given receptor - therefore triggering a weaker activation signal. In addition there even seems to be a threshold for the maximum speed that can be perceived by a given visual system. A very high velocity object, such as a spinning airplane propeller may appear to be entirely transparent, with only a slight "dimming" of the background signifying their presence.

To summarize: If time is a property of the object being observed and is not represented in the network as a property of the connections between nodes, then time is perceived and encoded as a pattern in the network and not as a value. This makes time a relation between an observed object's speed/rate of movement and an observer's perception/processing rate.

Based on initial observations, the predictor model had to address the following issues:

1. Feedback loops can create uncontrolled activity and irrecoverable states.
2. One time-step is not sufficient for accurate long-term predictions. A short-term, working memory should be used to extend the predictor.
3. Currently, every association is stored, leading to a quick saturation of the network. Competition and pattern decay should be introduced to:
  - a. Resolve conflict between opposing patterns
  - b. Decay old and rare patterns over time

## Results

### The Algorithm

The first version of a model based around the notion of a fixed-delay network used a very simple algorithm. At each time step, the currently firing network node is associated with the node that fired on the previous time step. The association is uni-directional, meaning that only the connection *from* the previous node is reinforced, while the connection *to* that node is not reinforced. The association weight is determined by the time that has passed since the previous node fired.

The initial implementation used a counter to determine the time that had passed between the firing of the two connected nodes. However, due to requiring counters for each node that fires during a single time step, the revised implementations instead use an extension of the action-potential charge of each node. While traditional artificial neural networks use a binary charge state of either 1 or 0, this extended model still outputs a binary energy value, but instead of returning to 0 immediately, the value is gradually decreased over a series of time steps. This made it possible to use the *charge falloff* as an individual measure of time for each node.

Below is the pseudo code for the algorithm used:

For each node **A** in the network

```
//Update CURRENT and PREDICTED charge:
•Retrieve charge from previous time step
•Decay the previous charge
•Add external stimuli to charge

•Calculate internal stimuli
  •For each node B (that is not A) that
    fired on the previous time step (in
    fired list)
    •If previous node B is connected to
      current node A
      •Calculate and add the input
        charge from B to A (node B's
        output & weight from B to A)
      •Decay the connection weight from
        A to B (if a connection exists)

  •Update current charge for node A
    (previous charge + external stimuli)
  •Update the predicted charge for node
    A (internal stimuli / past effect
    scaling value)

//Fire Nodes and Update Associations:
•If current charge > threshold
  •For each node B (that is not A) that
    fired on the previous time step
    •Increase the weight from B to A
    •Decrease the weight from A to B (if
      a connection exists)
  •Add the current node A to the fired
    list
  •Remove the oldest element from the
    fired list
```

The model showed the capability of reproducing time series patterns, albeit without any intermittent pauses between node activations. The pattern *sequence* was reproduced faithfully, but the reproduction was sped up with each node activating exactly one time-step after the previous one.

Another issue was that pattern loops could end up creating feedback patterns similar to Conway's "Game of Life" (Gardner, 1970).

### Separating Internal Activity from Prediction

The first change from the previous model was that the *predicted activity* was separated from the *actual activity* in the network. Instead of adding the internal activation energy to the total *charge* of each node (which determines whether the node would fire), it is instead accumulated in a new node property, the *prediction*. In the visualization that was used in our simulation, the *prediction* and the *charge* are displayed as two separate coloured bars to make it clear to the observer which was the *charge* caused by external stimulus and which was the predicted path. Separating the two immediately resolved the internal feedback problem of course. Figure 2 shows the implementation of the temporal pattern predictor which visualizes the separation between sensory input and prediction. The left most active threshold device is currently active. The two paths emanating from it are two previously perceived patterns of node activity. The darker shading indicates that a predicted pattern has been perceived more often and/or more recently than the weaker prediction.

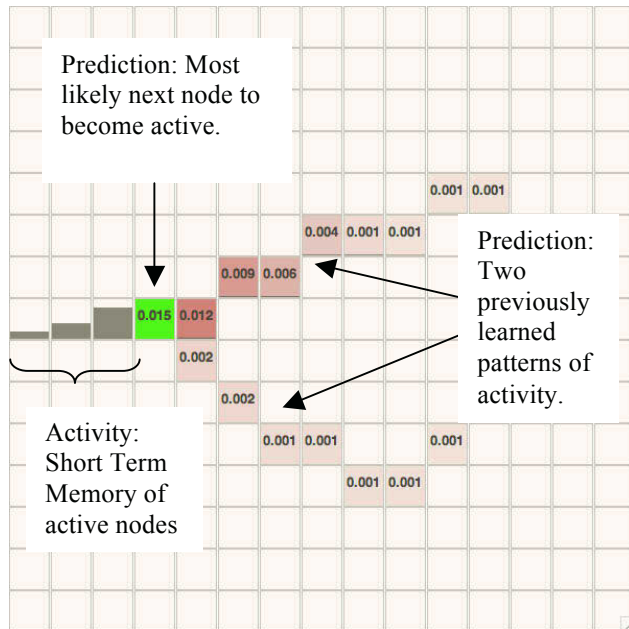


Figure 2 The Predictor model implementation

### Short Term Memory Trajectory

The main problem that occurred with the previous version of the predictor was that of pattern interference. In a network, any given node may be part of several trained patterns. When this node is activated, how does the network *choose* which

pattern to activate? While any of the patterns are valid when taking into account only the current state of the network, viewed as a series of states in *time*, the idea of all being valid becomes less likely. To illustrate this, the diagram in Figure 3 shows an experiment that sees four different opposing stimulus sequences presented to the network. Each of the four sequences passes through the same middle node and sequence pairs 1&2 and 3&4 share the same path. Figure 4 shows the false predictions the current model makes.

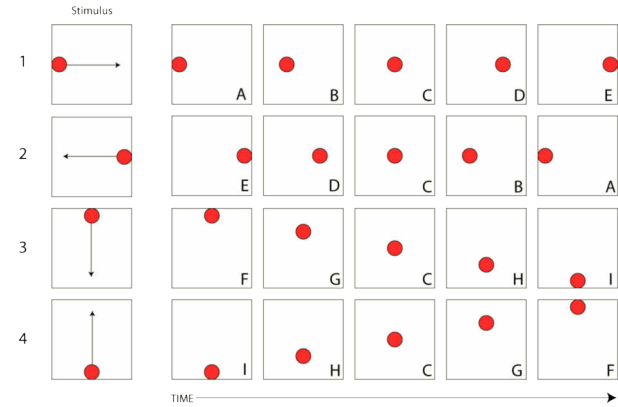


Figure 3 Experiment setup to test pattern interference

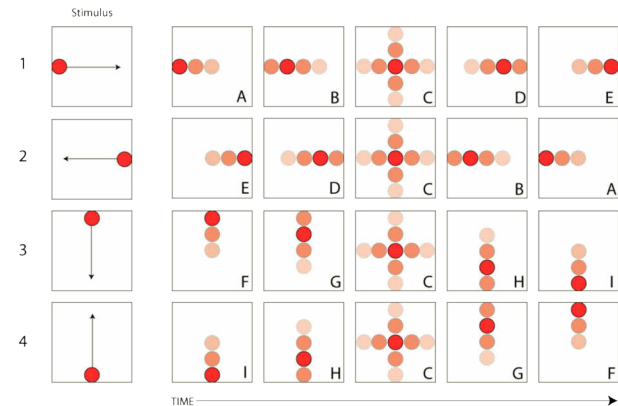


Figure 4 The false predictions the network makes due to pattern interference

Note how only 9 unique states can be identified from the perspective of this network. Viewing state C without taking the preceding time-steps into account, the predictor could validly predict either state D,B,H or G as the next possible position in the sequence.

In order to deal with this problem of differentiating between different temporal patterns, a short-term memory was introduced into the model. This sees a series of *past devices* associated to the current device under the notion that they are precursors of the currently active device. The further in the past these devices are active, the weaker the association to the current device. Figure 5 is an illustration of this mechanism. It shows an example of impact of short-term memory the possible predicted paths A, B and C. While all three are equally likely from the perspective of the current node, the

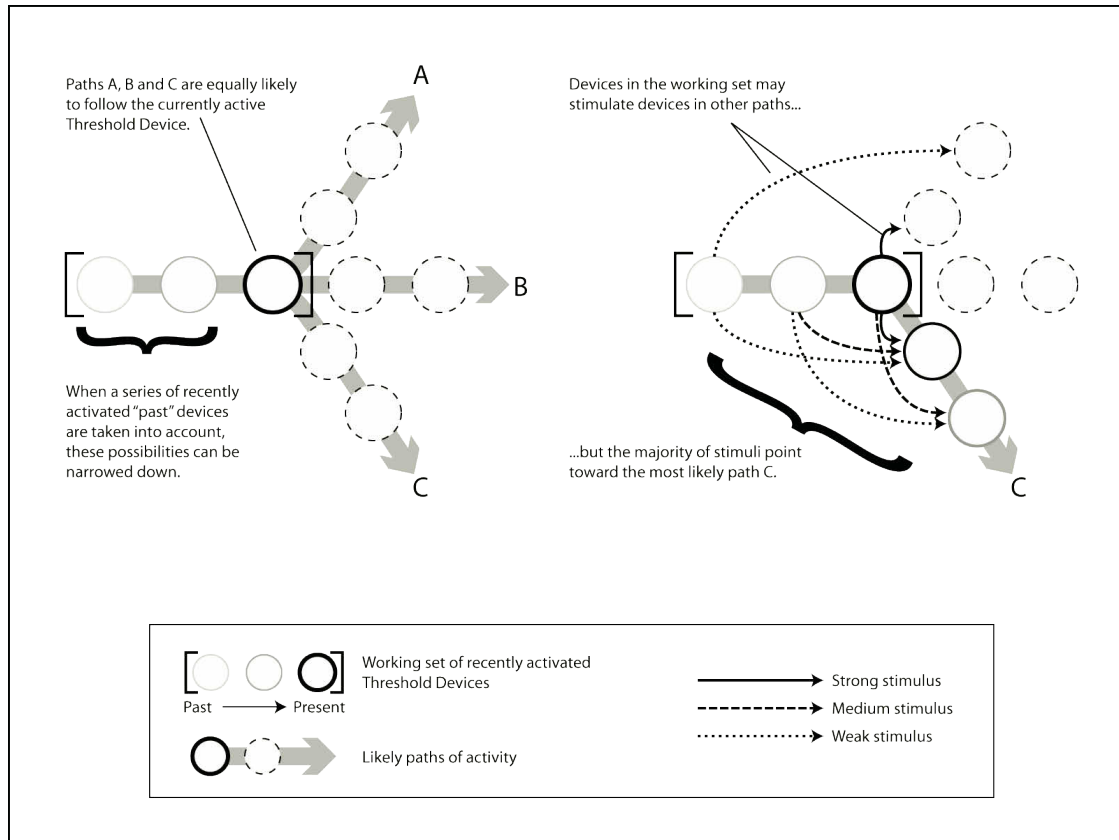


Figure 5 Diagram illustrating how the short-term memory allows the network to differentiate between predicted patterns.

additional stimulus from the two previous nodes in the short-term memory accumulate to support path C as the most likely pattern.

The result of adding a short-term memory of past nodes to the model is illustrated in Figure 6. The *perception* of the predictor has changed as it keeps the series of past nodes in mind. From the outset, this allows the network to differentiate between 20 unique states instead of just 9.

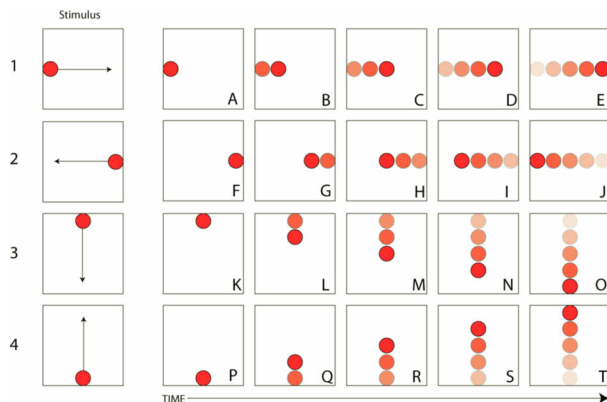


Figure 6 Introducing a short-term memory of active nodes changes the perception of the network

Based on the altered perception, Figure 7 shows that these 20 states are then associated with 20 different predictions. Since the influence from the short-term memory adds additional activity to the network, the problem of saturating the entire network becomes relevant. To deal with this problem the model needs to include the notion that certain patterns are *competitive* in that they represent opposing positions to a fact.

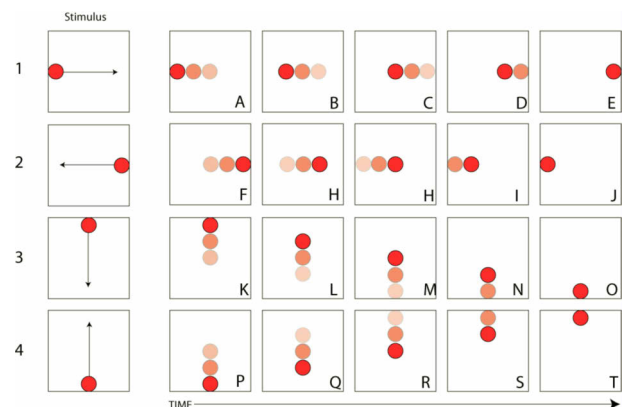


Figure 7 Using a short-term memory allows the network to differentiate between predictions

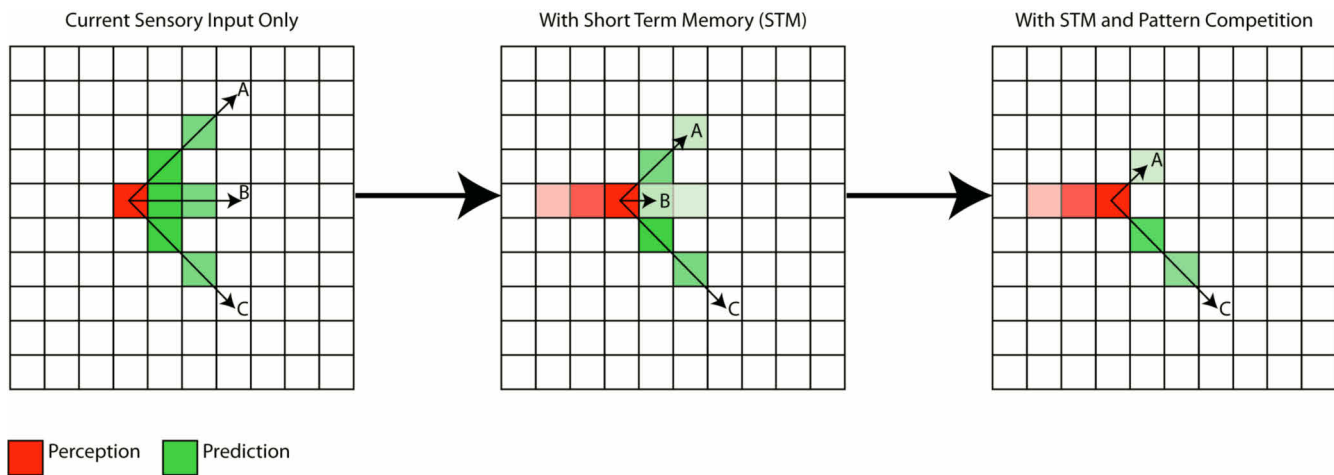


Figure 8 Pattern Inhibition enhances the prediction by inhibiting patterns only pointed to by individual nodes.

### Pattern Inhibition

A simple form of competition that was tested with this model was inhibition between patterns. Inspired by the lateral inhibition algorithm commonly used in edge-enhancement algorithms in computer vision, this mechanism allows currently active nodes and the nodes in short-term memory to inhibit all the nodes that they are currently not connected to. The result is a cumulative effect of inhibition. In parallel with exciting connected nodes as depicted in Figure 5, past and present nodes will inhibit every other node in the network, thus *enhancing* the strongest mutual patterns among them. Figure 8 illustrates this effect.

### Discussion and Further work

Since the model uses fixed delays, the predicted sequence is triggered simultaneously. Speeding up a predicted pattern is certainly desirable according to Braitenberg, who states that a predictive brain would need to “reproduce sequences at a more rapid pace” (Braitenberg, 1984, pg. 72). However, the interval information between the activation of nodes is lost, which contradicts Braitenberg’s earlier statement that “we implicitly assumed that the Ergotrix wires would be trained to reproduce sequences of activation at the same pace as the original occurrence of the sequences of events.” (Braitenberg, 1984, pg. 72).

Overcoming this issue would require re-introducing some form of internal self-activation to the network, albeit with a mechanism for preventing undesirable states such as feedback loops.

While this internal dynamic is desirable in a future model of the predictor mechanism, a feedback-free predictor has the benefit of stability and the ability to clearly visualize the discrepancy between sensed input and predicted output. It might also be possible to derive the timings from the prediction value of nodes.

### Further Work

Testing the predictor in an embedded scenario is the primary priority at this point. Two experiments are currently being prepared. The following is a summary of some of the early findings.

The scenario for both experiments is a chase-and-catch setup, with the goal being to stay as close as possible to a moving target. The simulation features a differential drive-driven agent with two distance sensors, which can be set to track a target. The aim of these experiments is to see whether adding the ability to anticipate the path of a moving target leads to more optimal behaviour. Note that in both experiments, the tracked target is moving *faster* than the agent. To catch the target, the agent therefore needs to intercept the target:

The first experiment implements the predictor as a probabilistic occupancy map (POM), inspired by previous work by Damian Isla (2002a, 2002b). The experiment sees the predictor network represent the location of a tracked object on an occupancy grid. The predictions generated therefore propose the possible future location of the tracked object and an agent’s sensors can be set to track the prediction instead of the ‘actual’ tracked object. This can be implemented using the current predictor, which only tracks a single point (the location of the object on the map).

Figure 9 shows the path of the agent without using the predicted position of the target, indicated by a cross. Figure 10 shows that the agent manages to get closer to the target when using the predicted position as the input. While this early result shows that our predictor can function as a POM, the result is still highly dependent on tuning. Further Increasing the speed of the target, would require an agent that is capable of adopting a strategy that does not involve it following the target, but instead waiting for it at an expected location. The current predictor would need to be extended to allow for such behaviour.

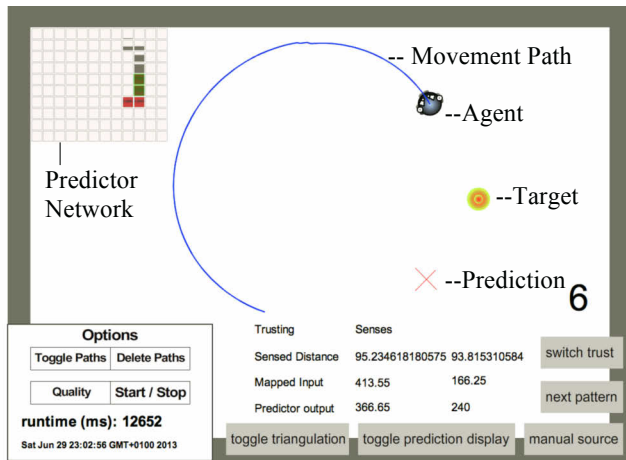


Figure 9 Chase-scenario with a moving target. The objective is to stay as close to the moving target as possible.

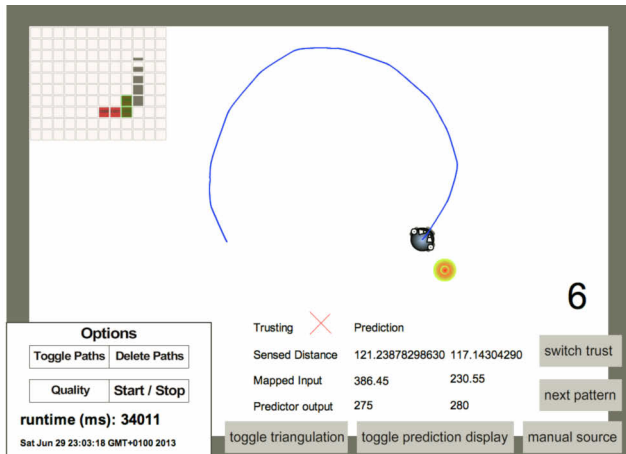


Figure 10 Using the predicted position of the target improves the behaviour.

Figure 11 shows the second experiment, which uses the predictor to generate a multi-dimensional sensory space. By using multiple individual predictors, one for each sensor input, separate predictions of the expected future state of each sensor are generated. In this case, the sensors used are two distance sensors. To function with the current predictor model, the distance reading need to be converted into discrete values that can be mapped to the predictor grid. Thus finding a suitable approach to balancing the performance and the accuracy of the predictions will be of particular interest.

The two distance sensor readings are not sufficient to accurately triangulate and predict the position of an object's position, since the distance sensors are omni-directional and always return a positive distance measurement in our simulation. The agent therefore requires an additional internal calculation that tells it in which direction (in front or behind) the observed object is. Combining this with the sensor readings thus gives discrete values ranging from negative

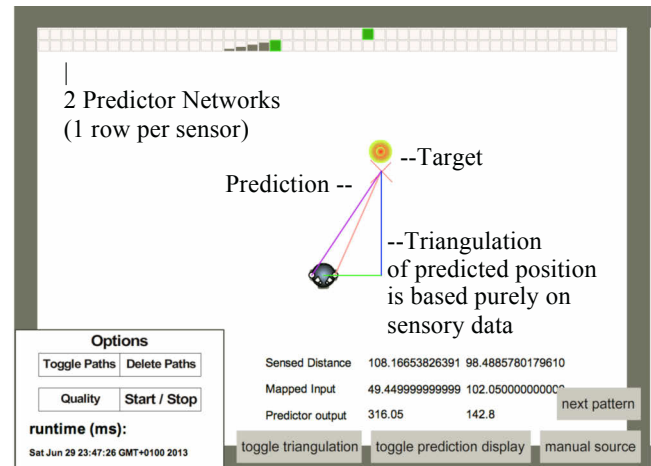


Figure 11 Experiment testing the use of the predictor network to anticipate sensory data. The future position of the target is triangulated and displayed.

(behind) to positive (in front) that can be fed into the predictor network. While this model already works for a stationary agent, the main issue that remains is effect of self-movement on the sensed values. This feedback-loop between the relative position of the agent with regard to the tracked object has to take into account, or rather counteract, the effect that self-movement has on the sensory data. Without this additional system in place, the current prototype will only work in a stationary agent, observing and predicting the location of a moving tracked object using its distance sensors.

## Conclusions

Our model of a Braitenberg-inspired temporal pattern predictor can successfully predict and visualize the path of a moving object and can avoid interference between crossing patterns through the use of short-term memory. A set of experiments successfully tested the model in the context of a chase scenario and has revealed several ways in which the current model could be further improved.

In the first experiment, the choice between following the sensed position of the target versus the predicted position on the occupancy map was controlled by the experiment. Using the user interface we were able to switch between the two and compare the resulting behaviours of the agent. A central topic for further research on our particular predictor model is the inclusion of an automatic switching mechanism that enables the agent to make a choice between purely reactive behaviour (following the sensory input directly) and pro-active behaviour (following the internal representation of the target on the POM). This in turn could be extended to include the ability to optimise the prediction mechanism by re-enforcing accurate predictions. Braitenberg's original description of the predictor includes this functionality and suggests a method of positive reinforcement based on classical conditioning (Pavlov, 1903).

The current model only supports sequences of single nodes. It only allows a single *past node* to be associate with a single



*current node*. To allow for more complex patterns and a wider range of application, it should be possible to associate groups of *past nodes* with groups of *current nodes*. The ability to support multi-point patterns could improve the predictor further. Connecting the currently separate sensor readings to the same network should give the predictor more evidence to base individual predictions on. As we saw with the inclusion of short-term memory, this could potentially improve its ability to differentiate between similar patterns.

**Acknowledgments.** This work was supported by the University of Brighton (UoB) Cultural Informatics Research Group, the UoB School of Computing, Engineering and Mathematics and the UoB Doctoral College Research Student Conference Support Grant.

## References

- Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*, The MIT Press. Cambridge, MA, USA
- Bryson, A. E. Ho, Yu-Chi (1969). *Applied optimal control: optimization, estimation, and control*. Blaisdell Publishing Company or Xerox College Publishing. p. 481.
- Gardner, Martin (1970-10). *Mathematical Games - The fantastic combinations of John Conway's new solitaire game "life"*. Scientific American 223. pp. 120–123. ISBN 0-89454-001-7.
- Hebb, D. O. (1949). *The Organization of Behavior*, Wiley and Son.
- Isla, D. and Blumberg, B., (2002a). *Object Persistence for Synthetic Creatures*, Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS): Part 3, ACM, New York, NY, USA: 1356-1363
- Isla, D. and Blumberg, B., (2002b). *New challenges for character-based ai for games*
- McCulloch, W. and Pitts, W. (1943). *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematical Biology 5(4): 115-133
- Pavlov, I. P. (1903) *The Experimental Psychology and Psychopathology of Animals From Nobel Lectures*, Physiology or Medicine 1901-1921, Elsevier Publishing Company, Amsterdam, 1967

# Evolution of Tail-Call Optimization in a Population of Self-Hosting Compilers

Lance R. Williams<sup>1</sup>

<sup>1</sup>Department of Computer Science, University of New Mexico, Albuquerque, NM 87131  
williams@cs.unm.edu

## Abstract

We demonstrate the evolution of a more complex and more efficient self-replicating computer program from a less complex and less efficient ancestor. Both programs, which employ a novel method of self-replication based on compiling their own source code, are significantly more complex than programs which reproduce by copying themselves, and which have only exhibited evolution of degenerate methods of self-replication.

## Introduction

Among living organisms, which employ many and varied mechanisms in the process of reproduction, examples of evolved mechanisms which are both more complex and more efficient than ancestral mechanisms, abound. Yet, nearly twenty years after (Ray, 1994)'s groundbreaking work on the Tierra system, in which the evolution of many novel (but degenerate) methods of self-replication was first demonstrated, there is still no example of a more complex and more efficient self-replicating computer program evolving from a less complex and less efficient ancestor.

This is not to say that there has been no progress in the field of artificial life since Tierra. Nor are we suggesting that increased reproductive efficiency is the only evolutionary path to increased complexity. The evolution of self-replicating programs of increased complexity has been demonstrated many times (Koza, 1994; Taylor and Hallam, 1997; Spector and Robinson, 2002), and perhaps most convincingly in the Avida system (Adami et al., 1994). However, more complex programs evolved in Avida only because complexity was artificially equated with efficiency in the sense that programs which learned to solve problems unrelated to self-replication were rewarded with larger rations of CPU time. No program in Avida (or in any other system known to us) has ever evolved a method of self-replication that is both more complex and more efficient than the method employed by its ancestor.

## A New Kind of Artificial Organism

Self-replicating programs have been written in both high-level languages and machine languages. We define a ma-

chine language program to be *interesting* if it prints a string at least as long as itself and halts when executed, and observe that the Kolmogorov complexity of interesting programs is lower than that of random strings of similar length. Now, if we were to train an adaptive compression algorithm on a large set of interesting programs, then the compressed programs which result would *look* like random strings. However, by virtue of being shorter, they would be more numerous relative to truly random strings of similar length. It follows that compression, which *decreases* redundancy by replacing recurring sequences of instructions with invented names, *increases* the density of interesting programs.

Since both processes increase redundancy and output machine language programs, it is natural to identify *decompression* with *compilation*, which increases redundancy by repeatedly generating similar sequences of instructions while traversing a parse tree. Viewed this way, programs written in (more expressive) high-level languages are compressed machine language programs, and compiling is the process of decompressing source code strings into object code strings which can be executed by a CPU.

If the density of interesting programs increases with the expressiveness of the language in which they are encoded (as the above strongly suggests), then one should use the most expressive language possible for any process, like genetic programming, which involves searching the space of interesting programs. However, if the goal is building artificial organisms, then high-level languages have a very serious drawback when compared to machine language. Namely, programs in high-level languages must be compiled into machine language before they can be executed by a CPU or be reified as a *distributed virtual machine* (Williams, 2012).

Given that we want our self-replicating programs to be both (potentially) reifiable and to evolve into programs of greater complexity and efficiency, we must ask: How can the advantages which derive from the use of a high-level language for genetic programming be reconciled with the fact that only machine language programs can be reified?

To address this question, we introduce a new and significantly more complex kind of artificial organism—a ma-

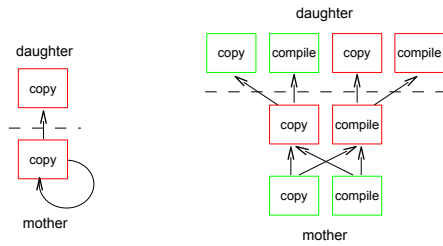


Figure 1: Conventional self-replicating program (left) copies itself by exploiting program-data equivalence of von Neumann architecture. Compiling quine self-replicating program (right) with source code genotype (green) and object code phenotype (red). Because the shortest correct implementation of copy is optimal, only the compiling quine is capable of non-degenerate evolution.

chine language program which reproduces by compiling its own source-code. See Figure 1. Conventional self-replicating programs reproduce by copying themselves. Optimum copiers accomplish this in time proportional to their length, and it is not very hard to write a copier which is optimum in this sense (or for one to evolve). It follows that shorter implementations are always more efficient, which leads to degenerate evolution, absent factors beyond efficiency. The possible variation in the implementation of a compiler is far larger. Even if the definition of the object language is stipulated, there is still a huge space of alternative implementations, including the syntax and semantics of the source language, the ordering of the decision tree performing syntactic analysis, and the presence (or absence) and effectiveness of any object code optimizing procedures.

In this paper we describe a machine language program which reproduces by compiling its own source code and use genetic programming to demonstrate its capacity for non-degenerate evolution. In the process we address questions such as: How can a program like a compiler, which implements a complex prescribed transformation, evolve improvements while avoiding non-functional intermediate forms? How can two lexically scoped programs be combined by crossover without breaking the product? How can a more efficient self-replicating program evolve from a less efficient ancestor when all mutations initially yield higher self-replication cost?

### A Simple Programming Language

Because a self-hosting compiler compiles the same language it is written in, it can compile itself. The language we used to construct our self-hosting compiler is a pure functional subset of Scheme which we call *Skeme*. Because it is purely functional, *define*, which associates values with names in a global environment using mutation, and *letrec*, which also uses mutation, have been excluded. The global environment itself is eliminated by making primitive functions constants. For simplicity, closures are restricted to one argu-

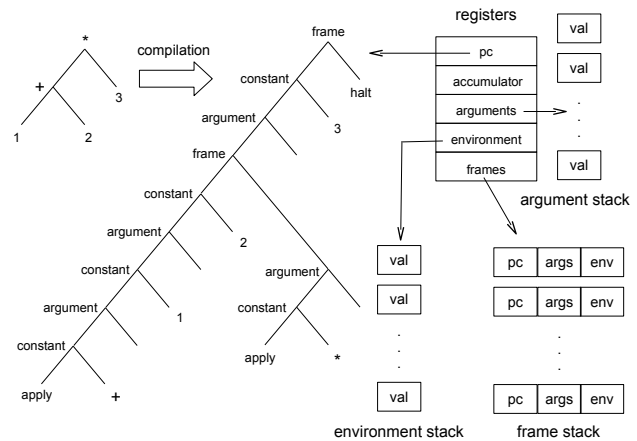


Figure 2: Virtual machine for evaluating compiled Scheme expressions showing its registers and associated heap-allocated data structures (Dybvig, 1987).

ment; user defined functions with more than one argument must be written in a curried style. This simplifies the representation of the lexical environment which is used at run-time by making all variable references integer offsets into a flat environment stack; these are termed *de Bruijn indices* and can be used instead of symbols to represent bound variables (De Bruijn, 1972).

One feature peculiar to Skeme is the special-form, *lambda+*. When a closure is created by *lambda+*, the closure's address is added to the front of the enclosed environment; the de Bruijn index for this address can then be used for recursive function calls. For example, the following function computes factorial:

```
(lambda+ (if (= %0 0) 1 (* %0 (%1 (- %0 1)))))
```

where *%0* is a reference to the closure's argument and *%1* is a reference to the closure's address.

### Tail-Call Optimization

Because the very first self-hosting compiler was written in Lisp, it is not surprising that it is possible (by including primitive functions which construct bytecode types) to write a very small self-hosting compiler in Skeme. See Figures 2 and 3.

The cost of compiling a given source code depends not only on its size, but also on the complexity of the source language, the efficiency of the compiler, and the cost of any object code optimizations it performs. Common compiler optimizations include constant folding, loop unrolling, function inlining, loop-invariant code motion, elimination of common subexpressions, and dead code elimination. Since a self-hosting compiler compiles *itself*, the efficiency of the object code it *generates* also affects compilation cost; it follows that minimizing the cost of self-compilation involves a complex set of tradeoffs. The most important of these is that object code optimizations must pay for themselves by yielding an increase in object code efficiency large enough

to offset the additional cost of compiling the source code implementing the optimization.

Most of the overhead associated with a function call involves the saving and restoration of evaluation contexts. In Skeme, these operations are performed by the *frame* and *return* bytecodes which push and pop the frame stack. However, when one function calls another function in a *tail position*, there is no need to save an evaluation context, because the restored context will just be discarded when the first function returns. A compiler which performs *tail-call optimization* recognizes when a function is called in a tail position and does not generate the code which saves and restores evaluation contexts. This not only saves time, it also saves space, since tail recursive function calls will not increase the size of the frame stack at runtime.

### A Quine which Compiles Itself

A *quine* is a program which prints itself. It is possible to write a quine in any programming language but Skeme's list-based syntax makes it possible to write especially short and simple quines. For example, in the following Skeme quine, an expression  $(\text{lambda } (\text{list } \%0 (\text{list quote } \%0)))$  which evaluates to a closure which appends a value to the same value quoted is applied to the same expression quoted:

```
((lambda (list %0 (list quote %0)))
 (quote (lambda (list %0 (list quote %0)))))
```

It is possible to define an expression  $\phi$  in Skeme which can compile any Skeme expression. The expression  $\phi$  evaluates to a curried function which takes a compiled expression and an uncompiled expression as arguments. The compiled expression is a *continuation*; the uncompiled expression is the source code to be compiled; applying the curried function to the halt bytecode yields a function which can compile top-level expressions. Inserting a copy of  $(\phi \text{ (make-halt)})$  into the unquoted half of the quine so that it compiles its result (and mirroring this change in the quoted half) yields

```
((lambda ((phi (make-halt))
          (list %0 (list quote %0))))
 (quote (lambda ((phi (make-halt))
          (list %0 (list quote %0)))))
```

which, although not a quine itself, returns a quine when evaluated. Significantly, this quine is not a source code fixed-point of the Skeme interpreter but an object code fixed-point of Dybvig's virtual machine. In effect, it is a quine in a low-level language (phenotype) which reproduces by compiling a compressed self-description written in a high-level language (genotype).

In prior work on evolution of self-replicating programs there has been no distinction between phenotype and genotype; mutations are made on the same representation which is evaluated for fitness. In contrast, in living organisms, small changes in genotype due to mutation can be amplified by a development process and result in large changes in phenotype; it is phenotype which is then evaluated for fitness. In

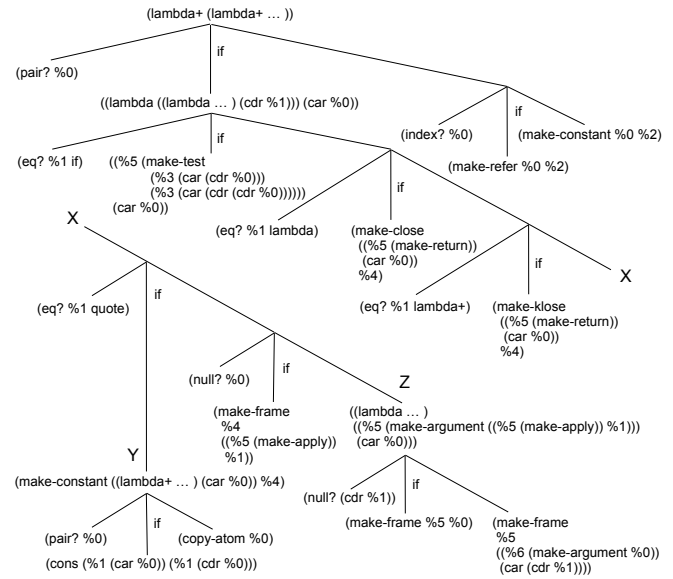


Figure 3: An expression  $\phi$  for compiling Skeme into object code able to compile itself. The X indicates a break in the figure; the subtree labeled Y copies the Skeme source code and the subtree labeled Z compiles function applications.

a compiling quine, small changes in source code (genotype) are amplified by compilation (development) yielding much larger changes in object code (phenotype) and it is object code which determines fitness, since its execution consumes the physical resources of space and time.

### Related Work

(Stephenson et al., 2003) described a genetic programming system which learns priority functions for compiler optimizations including hyperblock selection, register allocation, and data prefetching. (D'haeseleer, 1994) described and experimentally evaluated a method for context preserving crossover. (Kirshenbaum, 2000) demonstrated a genetic programming system where crossover is defined so that it respects the meaning of statically defined local variables.

Several authors have explored the idea of staged or alternating fitness functions. (Koza et al., 1999) used a staged fitness function as a method for multi-objective optimization. (Pujol, 1999) described a system where the fitness function is switched after a correct solution is discovered to a function which minimizes solution size. (Zou and Lung, 2004) and (Offman et al., 2008) used alternating fitness functions to preserve diversity in genetic algorithm derived solutions to problems in water quality model calibration and protein model selection.

### Genetic Programming

Our approach to genetic programming is motivated by the fact that gene duplication followed by specialization of one or both copies is a common route to increased complexity in biological evolution (Finnigan et al., 2012). We introduce

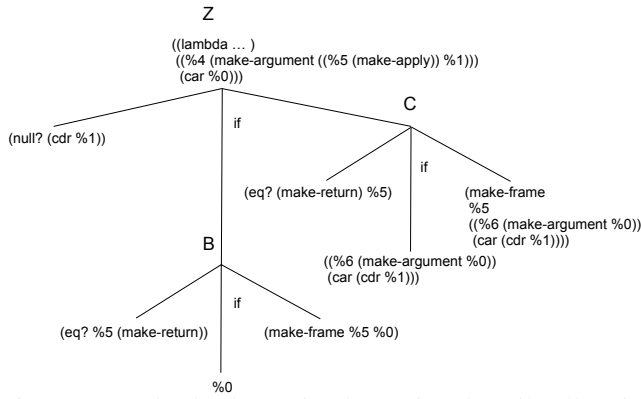


Figure 4: Evolved subtrees implementing the tail-call optimizations which characterize the B and C genotypes. The A genotype performs neither optimization while the D genotype performs both. Both optimizations check to see if the continuation is a return bytecode, which performs a frame stack pop. If so, the push-pop sequence is not generated, resulting in significant savings in time and space usage.

two mutation operators called *bloat* and *shrink* which play roles analogous to gene duplication and specialization and employ these in a genetic programming system where fitness alternates between object code based definitions of complexity and self-replication efficiency. In teleological terms, the bloat operator attempts to increase complexity by adding source code while the shrink operator attempts to increase self-replication efficiency by removing it.

### Alternating Fitness Function

Time is divided into ten generation periods termed *epochs* which alternate between two types, *flush* and *lean*. In flush epochs, fitness is defined as *effective complexity* while in lean epochs it is defined as *self-replication efficiency*.

A test bytecode is defined to be *non-trivial* if both of its continuations are exercised in the course of self-replication. This will only happen if the predicate expression in the *if* special-form from which the test bytecode is compiled sometimes evaluates to true and sometimes to false. The number of non-trivial test bytecodes in the object code is a good measure of the source code's effective complexity. Consequently, in flush epochs the number of non-trivial test bytecodes in the object code is *maximized*.

Because frame stack pushes and pops are the most expensive operation performed by the virtual machine, they are an excellent proxy for overall self-replication cost. Consequently, in lean epochs, the number of frame stack pops, which are implemented by the return bytecode, is *minimized*.

Mutations can be classified as beneficial, neutral, harmful, and lethal. The purpose of the bloat operator is to introduce source code which can be shaped by the shrink operator and by crossover. Significantly, the introduced code does not change the value of any expression which contains

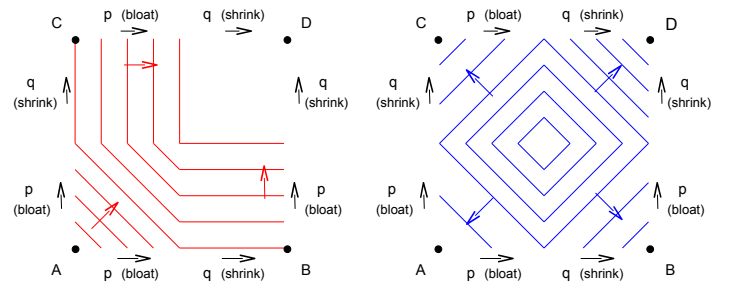


Figure 5: Contour plots of fitness landscapes during flush (left) and lean (right) epochs. Colored arrows point in directions of increased fitness. In lean epochs, the four genotypes A, B, C, and D occupy islands separated by valleys of decreased fitness; the bloat mutations necessary for A to evolve into any of the other genotypes are harmful since they increase the cost of self-replication. In contrast, the shrink mutations required for A to evolve into any of the other genotypes are beneficial. In flush epochs, the situation is reversed—the bloat mutations are beneficial and the shrink mutations are harmful since they increase and decrease effective complexity respectively. Alternating between the two fitness functions creates paths between the A and D genotypes consisting solely of beneficial mutations.

it; it is *value-neutral* with respect to evaluation. Because (by their nature) they increase the cost of self-replication without breaking the compiler, bloat mutations (although never lethal) are harmful during lean epochs.

In contrast, shrink mutations are beneficial when they reverse bloat mutations during lean epochs and can be harmful when they reverse bloat mutations during flush epochs. However, shrink mutations have two different and more pronounced effects. First, a shrink mutation can remove code and break the compiler, in which case it is lethal. Second, it can shape the result of a bloat mutation in a way which decreases the cost of self-replication, in which case it will be strongly beneficial during lean epochs and become fixed in the population.

### Bloat

The source code for the self-hosting compiler contains boolean-valued expressions with six different syntactic forms. Excluding primitive functions, the source code contains six different expressions of constant value. A random syntactic form can be combined with a random de Bruijn index and (if necessary), a random constant-valued expression, to construct a random boolean-valued expression,  $\phi$ .

The bloat operator is defined by five rules. The first four rules define a recursive procedure which applies the bloat operator in selected contexts. The last rule replaces a function application with an *if* expression which returns the same value regardless of whether a random boolean-valued expression,  $\phi$ , evaluates to true or false. Consequently, the



value of the expression is the same before and after the mutation. The fact that the bloat operator is value-neutral with respect to evaluation is important because only viable individuals (those which correctly self-replicate) are copied to the next generation; and although a bloat mutation typically introduces expressions which are not evaluated during self-replication (which greatly reduces the fitness of affected individuals by increasing their self-replication costs) affected individuals always remain viable because bloat mutations cannot actually break the compiler which contains them. The five rules which define the bloat operator are

1.  $(\text{lambda}[+] e_1) \rightarrow (\text{lambda}[+] e'_1)$
2.  $((\text{lambda}[+] e_1) e_2) \rightarrow ((\text{lambda}[+] e'_1) e'_2)$
3.  $(\text{if } e_1 (\text{id } e_2) e_3) \rightarrow (\text{if } e_1 (\text{id } e_2) e_3)$
4.  $(\text{if } e_1 e_2 e_3) \rightarrow (\text{if } e_1 e'_2 e'_3)$
5.  $(f e_1 \dots e_N) \rightarrow (f e_1 \dots e_N) \parallel (\text{if } \phi (\text{id } (f e_1 \dots e_N)) (f e_1 \dots e_N))$

where  $f$  is a primitive function,  $\phi$  is a random boolean-valued expression,  $\text{id}$  is the identity function, and primes mark expressions which are recursively expanded. Alternative right hand sides are separated by vertical bars; the alternative to the left of the  $\parallel$  (no mutation) is chosen with 95% probability; the remaining alternative (mutation) is chosen otherwise. The identity function serves as a value neutral tag in a meta-syntax; because the third rule has the same left and right hand sides, the recursive procedure which applies the bloat operator will not descend into  $\text{id}$  subtrees marked with this tag; this prevents the compounding of bloat mutations.

### Shrink

The rules defining the shrink operator serve two purposes. the first purpose is to reverse mutations introduced by the bloat operator; the fourth shrink rule removes the tagged  $\text{id}$  expressions generated by the bloat operator so that a bloat mutation followed by a shrink mutation (of this type) has no net effect. The second purpose is to simplify function applications; the last shrink rule replaces an expression where a function is applied to one or more values with just one of those values. Because these rules also remove the identity function tags inserted by the bloat operator, the expression which results from a shrink mutation is again subject to bloating. The five rules which define the shrink operator are

1.  $(\text{lambda}[+] e_1) \rightarrow (\text{lambda}[+] e'_1)$
2.  $((\text{lambda}[+] e_1) e_2) \rightarrow ((\text{lambda}[+] e'_1) e'_2)$
3.  $(\text{if } e_1 e_2 e_3) \rightarrow (\text{if } e_1 e'_2 e'_3)$
4.  $(\text{if } e_1 (\text{id } e_2) e_3) \rightarrow (\text{if } e_1 (\text{id } e'_2) e'_3) \parallel e_2 \mid e_3$
5.  $(f e_1 \dots e_N) \rightarrow (f e_1 \dots e_N) \parallel e_1 \mid \dots \mid e_N$

Table 1: Complexities and self-replication costs.

	A	B	C	D
non-trivial tests	8	9	9	10
returns	551	333	432	183

where  $f$  is a primitive function,  $\text{id}$  is the identity function, and primes mark expressions which are recursively expanded. Alternative right hand sides are separated by vertical bars; the alternative to the left of the  $\parallel$  (no mutation) is chosen with 95% probability; one of the remaining alternatives (mutation) is chosen otherwise (each with equal probability). Unlike the bloat operator, which is value neutral, the shrink operator changes the object code generated by the compiler when it modifies an expression which is evaluated during self-replication. In the case of the fourth shrink rule, this often reverses a harmful bloat mutation, in which case the shrink mutation is beneficial. However, in the case of the last shrink rule, the mutation most often breaks the compiler. Very rarely, the shrink mutation does not break the compiler but instead results in a decrease in self-replication cost.

The problem which plagues many genetic programming systems, in which code trees grow larger with increasing time, does not occur for two reasons. First, the use of the  $\text{id}$  function as a tag prevents the bloat operator from being applied within  $\text{id}$  expressions which were themselves just created. Second, the shrink operator reverses bloat mutations, and bloat mutations not yielding a decrease in self-replication cost are strongly selected against during lean epochs.

The combined effect on fitness of these two mutation operators is complex. After a pair of bloat and shrink mutations, a more complex source code must be analyzed by a more complex compiler, a change which might (but more likely will not) pay for itself by an increase in the efficiency of the generated object code.

### Crossover

Because the self-hosting compiler is a complex lexically scoped program, variables which are defined in one scope will not necessarily be defined in other scopes. If we employed the standard method of non-homologous crossover used in most work on genetic programming, then subtrees could be inserted into scopes where one or more variables are undefined, and this would break the compiler. We address this problem by employing the *homologous* crossover method described by (D'haeseleer, 1994). In this method, the crossover operator descends into both parent trees in parallel; points where the two parent trees differ are subject to crossover, with the child receiving the subtree of either parent with equal probability. D'haeseleer notes that homologous crossover facilitates convergence (fixation) since children resulting from the crossover of identical parents will also be identical to the parents.

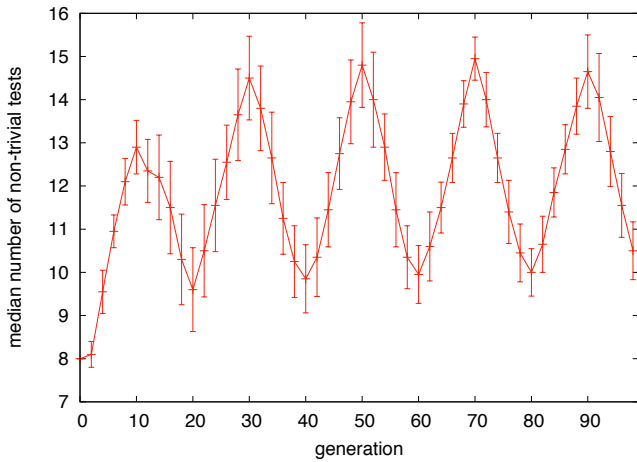


Figure 6: The median number (in a population of size 200) of non-trivial test bytecodes averaged over 20 runs (error bars show plus or minus one standard deviation). Because each non-trivial test bytecode results from a bloat mutation at a distinct point in the  $\varphi$  expression, this graph demonstrates that mutation is in no way restricted to the two points relevant to the evolution of tail-call optimization.

## Genotypes

Function applications involving one and two arguments are compiled at two different points in the  $\varphi$  expression and each of these points is a potential target for a pair of bloat and shrink mutations which would partially implement tail-call optimization. We call the genotype of programs which perform neither optimization *A*, one (or the other) optimization *B* (or *C*), and both optimizations, *D*. Both optimizations check to see if the continuation is a return bytecode, which performs a frame stack pop. If so, the push-pop sequence is not generated, resulting in significant time and space savings. See Figure 4. Lower bounds for the complexity and self-replication cost of each of the four genotypes are shown in Table 1. Finally, the relative fitnesses of the four genotypes are shown graphically, in the context of the fitness landscapes for the flush and lean epochs, in Figure 5.

## Experimental Results

The initial population consisted of two hundred identical individuals of genotype *A* at the beginning of a flush epoch (in which fitness is equated with effective complexity). In the first step of the genetic algorithm, the bloat and shrink operators are applied to all individuals in the population and the mutants which result are tested for viability. To test for viability, the mutant is evaluated to produce a daughter, and the daughter is evaluated to produce a granddaughter. The mutant is classified as viable if the daughter and granddaughter contain the same number (greater than zero) of bytecodes (this is done in lieu of a much more expensive test of actual structural equivalence). Viable mutants replace their pro-

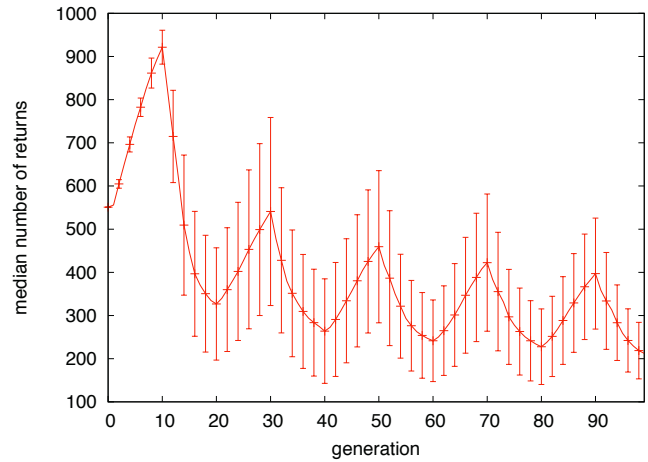


Figure 7: The median number (in a population of size 200) of return bytecodes executed during self-replication averaged over 20 runs (error bars show plus or minus one standard deviation).

genitors in the population.

The population is then subjected to crossover using tournament selection. In each tournament, four individuals are chosen at random (with replacement). The winners of two tournaments are then combined using crossover, and the resulting individual is tested for viability. The crossover operation is repeated until it yields two hundred viable individuals which comprise the population of the next generation.

The above process is repeated for nine more generations, then the epoch is switched to lean (in which fitness is equated with self-replication efficiency). The genetic algorithm is run for a total of 100 generations (five flush epochs interrupted by five lean epochs).

In an initial experiment, the system was run twenty times. The median number of interesting test bytecodes contained in the compiled  $\varphi$  expression and the median number of return bytecodes executed during self-replication were then plotted as a function of generation; see Figures 6 and 7. As expected, both complexity and self-replication cost increase in flush epochs and decrease in lean epochs. After 40 generations (two flush-lean cycles), the median complexity at the end of flush epochs is nearly double its initial value, which means that the majority of individuals contain 7 or more predicates which compile to non-trivial test bytecodes not present in the initial population. Furthermore, the median complexity at the end of lean epochs is always 10 or more, which suggests that either 1) the shrink operator is not fully able to reverse the effects of the bloat operator so that one or more bloat mutations (on average) survive through lean epochs; or 2) one (or both) of the *B* and *C* alleles is fixed in the population. Examination of Figure 7 shows that after 40 generations, the median self-replication cost at the end of lean epochs is slightly more than half of its initial value.

This is consistent with evolution of one or both of the B and C genotypes. Self-replication cost continues to increase and decrease (depending on epoch) eventually reaching a point where the median value at the end of the fifth lean epoch is nearly three times smaller than the initial value. This is consistent with the evolution of the D genotype.

After running the system 100 times, the probabilities of the B, C, and D genotypes evolving and for the mutations becoming fixed in the population were estimated. See Table 2. Notably, the most complex and most efficient genotype, D, evolved within 100 generations 81 times. Additionally, the average and median number of generations required for each genotype to evolve and for the mutations to become fixed were also estimated. Considering only the 81 runs in which the D genotype evolved, the average number of generations required was approximately 36 and the median number was 29.

Table 2: Generation of initial evolution and fixation.

	B	C	D	B'	C'	D'
probability	0.90	0.91	0.81	0.89	0.78	0.70
mean	21.8	24.5	35.8	29.9	34.3	43.3
std. dev.	21.0	22.0	24.5	21.1	22.2	24.5
median	11	13	29	17	33	36

If we know the average numbers of individuals of a given genotype in each generation, then we can compute cumulative distribution functions for evolution and fixation of that genotype; see Figure 8. If we examine the c.d.f.'s we see several interesting things.

First, the c.d.f.'s for evolution of genotypes have zero slope during lean epochs, which suggests that new genotypes typically appear during flush epochs, when fitness is equated with effective complexity. Conversely, the c.d.f.'s for genotype fixation have zero slope during flush epochs, which leads us to conclude that fixation of genotypes typically occurs during lean epochs, when fitness is equated with efficiency. This is consistent with an increase in diversity during flush epochs and a decrease during lean epochs.

Second, there is always a lag between the generations of evolution and fixation, and the size of the lag depends on the improvement in self-replication efficiency—the greater the improvement, the shorter the lag. The C allele (which confers an advantage of 119 returns relative to the A allele) requires more time for fixation than the B allele (which confers an advantage of 218 returns).

If we know the generation in which each genotype evolved, it is possible to estimate probabilities for each of the pathways leading from the (least complex and least efficient) A genotype to the (most complex and most efficient) D genotype; see Table 3. This analysis shows that in 64% of the runs in which D evolved, one of the B or C alleles evolved and was fixed prior to the evolution of the other;

Table 3: Probabilities of pathways to D genotype.

$t_B < t_C = t_D$	$t_C < t_B = t_D$	$t_B < t_C < t_D$	$t_C < t_B < t_D$
0.33	0.31	0.26	0.09

the D genotype then evolved by mutation from an ancestral program of the B or C genotype. However, in 35% of the runs in which D evolved, something (arguably) more interesting happened. Namely, the B and C alleles evolved in distinct lineages before either was fixed. The D genotype then evolved when an individual with the B allele and an individual with the C allele were combined by crossover. Stated differently, in 35% of the runs where D evolved, beneficial traits which evolved separately were combined by crossover to produce a child program more complex and more efficient than either parent program.

### Future Work

This paper describes work that, although preliminary, opens many avenues for further exploration, including

- Determining whether or not a self-replicating program which reproduces by compiling itself can evolve the optimum order for the tests comprising the decision tree which performs syntactic analysis; this would require a new mutation operator which can reorder nested-*if* expressions.
- Determining whether or not it is possible to evolve dead code elimination, which would be a useful optimization in a system which includes mutation operators (like bloat) which (in effect) introduce dead code; to accomplish this, the bloat operator would have to generate a much larger set of  $\phi$  expressions, including dereferencing source code with *car* and *cdr* combinations.
- In the present system, de Bruijn indices are used mainly to simplify the compilation process by eliminating the need for static analysis; however, it is difficult to see how new lexical scopes could evolve (via a new mutation operator which introduces *lambda* expressions) unless bound variables are represented by symbols, and this would mean that the self-hosting compiler must be generalized so that it performs static analysis.
- Demonstration of *auto-constructive evolution* as described by (Spector and Robinson, 2002), in which artificial organisms possess not only their own means of self-replication, but also of producing variation; this would require coding all mutation operators in Skeme and including this code in the subtree of the self-hosting compiler which copies quoted expressions.
- Reification of the compiling quine as a self-replicating distributed virtual machine (including the items listed above) and demonstration of evolution of increased complexity and self-replication efficiency by reified artificial organisms.

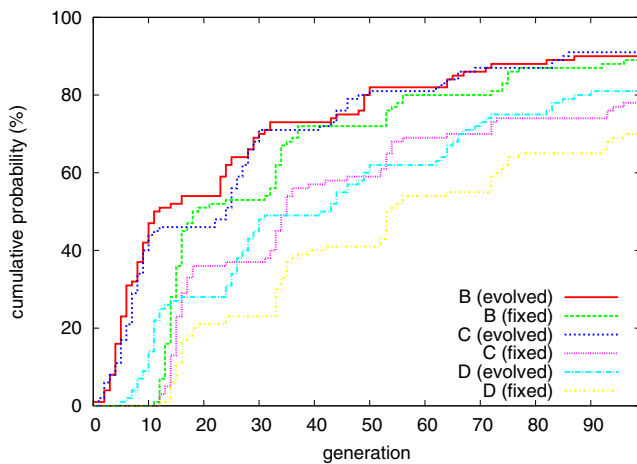


Figure 8: Cumulative distribution functions representing the probabilities that genotypes B, C, and D have evolved and are fixed by the given generation.

## Conclusion

We introduced a new type of self-replicating program which (unlike previous self-replicating programs) includes distinct phenotype and genotype components. Although the program is encoded in machine language, and (for this reason) can be executed on a CPU (or reified as a distributed virtual machine) it reproduces by compiling itself from its own source code, which is written in a more expressive high-level language. Because compiling is an intrinsically more complex process than copying, there is a much larger space of implementations to be explored by an evolutionary process; because its genotype is encoded in a high-level language, the space of neighboring self-replicating programs can be more efficiently probed.

To address the problem of how a complicated lexically scoped program like a compiler can evolve into a more complex and efficient program without breaking, we designed, implemented and tested a novel genetic programming system, which uses a pair of mutation operators analogous to gene duplication and specialization, together with homologous crossover and an alternating fitness function which selects for complexity or efficiency depending on epoch. Using this system, we experimentally demonstrated the evolution of several self-replicating programs of increased complexity and efficiency from a less complex and less efficient ancestor. We were able to show that in a population of 200 individuals, the most complex and efficient self-replicating program evolved within 100 generations in over three quarters of all trials, and by crossover of less complex and less efficient parent programs a significant fraction of the time.

## Acknowledgements

Thanks to Jeff Barnett, Stephanie Forrest, Ben Edwards, Neal Holtschulte and Melanie Moses for helpful comments.

## References

- Adami, C., Brown, C. T., and Kellogg, W. (1994). Evolutionary learning in the 2D artificial life system “Avida”. In *Artificial Life IV*, pages 377–381. MIT Press.
- De Bruijn, N. G. (1972). Lambda calculus notation with nameless dummies: a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Indagationes Mathematicae*, 34:381–392.
- D’haeseleer, P. (1994). Context preserving crossover in genetic programming. In *IEEE World Congress on Computational Intelligence*, pages 27–29.
- Dybvig, R. K. (1987). *Three implementation models for Scheme*. PhD thesis, University of North Carolina.
- Finnigan, G., Hanson-Smith, V., Stevens, T., and Thornton, J. W. (2012). Evolution of increased complexity in a molecular machine. *Nature*, 481(7381):360–364.
- Kirshenbaum, E. (2000). Genetic programming with statically scoped local variables. In *Genetic and Evolutionary Computation (GECCO)*.
- Koza, J. (1994). Artificial life: spontaneous emergence of self-replicating and evolutionary self-improving computer programs. In Langdon, C., editor, *Artificial Life III*, pages 225–262. Addison Wesley.
- Koza, J., Bennet, F., Andre, D., and Keene, M. (1999). The design of analog circuits by means of genetic programming. *Evolutionary Design by Computers*, pages 365–385.
- Offman, M. N., Tournier, A. L., and Bates, P. A. (2008). Alternating evolutionary pressure in a genetic algorithm facilitates protein model selection. *BMC Structural Biology*, 8(34).
- Pujol, J. C. F. (1999). *Evolution of artificial neural networks using a two-dimensional representation*. PhD thesis, School of Computer Science, University of Birmingham, UK.
- Ray, T. S. (1994). An evolutionary approach to synthetic biology, zen and the art of creating life. *Artificial Life*, 1:179–209.
- Spector, L. and Robinson, A. (2002). Genetic programming and auto-constructive evolution with the push programming language. *Genetic Programming and Evolvable Machines*, 3(1):7–40.
- Stephenson, M., Amarasinghe, S. P., Martin, M. C., and O’Reilly, U. (2003). Meta optimization: improving compiler heuristics with machine learning. In Cytron, R. and Gupta, R., editors, *PLDI*, pages 77–90. ACM.
- Taylor, T. and Hallam, J. (1997). Studying evolution with self-replicating computer programs. In *Fourth European Conf. on Artificial Life*, pages 550–559. MIT Press.
- Williams, L. (2012). Robust evaluation of expressions by distributed virtual machines. In *Unconventional Computation and Natural Computation (UCNC)*, Orleans, France.
- Zou, R. and Lung, W. (2004). Robust water quality model calibration using an alternating fitness genetic algorithm. *J. Water Resource Planning Management*, 130(6):471–479.



## Bladder Cancer Specific Pathway Interaction Networks

Qinxin Pan<sup>1</sup>, Ting Hu<sup>1,3</sup>, Angeline S. Andrew<sup>2</sup>, Margaret R. Karagas<sup>2</sup> and Jason H. Moore<sup>1,2,3</sup>

<sup>1</sup>Department of Genetics, Geisel School of Medicine, Dartmouth College, Hanover, NH 03755, USA

<sup>2</sup>Department of Community and Family Medicine, Geisel School of Medicine, Dartmouth College, Hanover, NH 03755, USA

<sup>3</sup>Institute for Quantitative Biomedical Sciences, Dartmouth College, Hanover, NH 03755, USA  
jason.h.moore@dartmouth.edu

### Abstract

With the development of high-throughput technologies, monitoring biological systems comprehensively has become feasible and affordable. However, the transition from high-throughput data to the underlying biology of various phenotypes remains challenging. Pathway analysis identifies biological processes that are associated with a particular phenotype, which provides insights into the underlying biological mechanisms. Therefore, pathway analysis has become a popular tool for analyzing high-throughput data. Most existing pathway analysis methods are based on a simple assumption that pathways act in isolation whereas they cooperate with each other in a complex manner. In this study, we focus on pathway interactions that are associated with bladder cancer risk. We identify disease-specific pathway-pathway interactions based on SNP-SNP interactions and gene-gene co-expression relationships. By analyzing the structure of pathway interaction networks, we highlight the “central” pathways that should be further studied.

### Introduction

Techniques such as high-throughput sequencing and gene/protein profiling have transformed biological research by enabling comprehensive monitoring of a biological system (Khatri et al., 2012). Conventional analyses of high-throughput data usually test the association of individual genes/proteins with a given phenotype. Although successfully applied in many studies, this approach fails to provide insights into the underlying biological mechanisms of the phenotype being studied. As an alternative approach, pathway analysis highlights the risk associated biological processes. The mechanisms of pathway associations can be used for developing strategies to diagnose, treat and prevent complex diseases (Ramanan et al., 2012), which makes high-throughput datasets more often viewed as a foundation to discover associated pathways (Hirschhorn, 2009).

In pathway analyses, gene sets corresponding to biological pathways are tested for significant associations with a phenotype. Multiple methods have been developed and among them pathway enrichment approach is the most popular one. Most pathway-enrichment-approach studies follow two categories: the threshold-based framework and the

rank-based framework. Threshold-based approaches usually statistically evaluate the fraction of genes in a particular pathway among all the significant markers (Boyle et al., 2004). Rank-based approaches rank all markers based on their significances and then look for pathways that have better rankings than the overall distribution (Subramanian et al., 2005). Although successfully applied in many studies, enrichment approaches depend on single-marker statistics and treat each gene independently. In reality, biological systems are driven by complex biomolecular interactions instead of individual genes (Schadt et al., 2009). Thus, methods that take biomolecular interactions into account remain needed.

Most pathway analysis approaches assume that each pathway is independent of the others, which could be problematic (Khatri et al., 2012). Pathways do not operate in isolation. Instead, they cooperate and work together as a unit. A recent study of yeast showed that rewiring of genetic interactions in response to DNA damage are more likely to occur among pairs of genes that belong to two different biological processes (Bandyopadhyay et al., 2010). Although whether interactions are more likely to occur within the same pathway or across different pathways in human is unknown, the significance of pathway interactions is in-negligible. There are several possible ways two pathways can interact: sharing components (Lu et al., 2007), components physically interacting with counterparts from the other pathway (Lu et al., 2007; Guo and Wang, 2009), components relating with components of the other pathway via transcription (Guo and Wang, 2009), etc. The different interacting ways can be reflected in different manners including pathway overlapping, direct or indirect protein protein interactions (PPIs), co-regulation of genes etc. So to fully understand pathway interactions, integration of knowledges from different levels is required.

Several studies have been done to address pathway interactions. Tong et al. carried out genetic screening and used synthetic lethality of two mutations to indicate interaction between two pathways (Tong et al., 2004). Later, Kelley et al. used such genetic interactions to link “pathways”, which is defined as sets of densely connected pro-



teins in the protein-protein-interaction (PPI) network (Kelley and Ideker, 2005). Meanwhile, overlapping in genes, proteins or metabolite contents between pathways has also been used to gain insights in possible relationships between pathways (Isserlin et al., 2010). Lastly, a function based approach is to find possible pathway crosstalks by looking at protein interactions between pathways (Li et al., 2008). Most previous methods highlight pathway crosstalks under general conditions. The crosstalks identified are not associated with any particular phenotype. In this study, we make use of the disease case-control data and search for the disease specific pathway interactions only, which distinguish our work from previous ones the most.

Meanwhile, network science has emerged as a useful tool for modeling biological interactions and dependencies (Ideker and Sharan, 2008). Compared to analysis of groups of distinct objects, networks provide more information on the relatedness and interconnectivity of them, which makes network a suitable framework for investigating pathway interactions.

In this study, we focus on pathway interactions that are associated with bladder cancer risk. We infer pathway interaction networks from two different levels: SNP-SNP interactions and gene-gene co-expression relationships. By analyzing the structure of pathway interaction networks, we identify essential pathways that hold great potentials for future studies.

## Methods

### SNP interaction network

A recent study characterized the space of pairwise interactions in a population-based bladder cancer association study (Hu et al., 2011). In their SNP interaction network, each vertex corresponds to a single nucleotide polymorphism (SNP). An edge linking a pair of vertices corresponds to an interaction between two SNPs. Weights assigned to each SNP and each pair of SNPs quantify how much of the disease status the corresponding SNP and SNP pair can explain. The significance of the SNP interaction network is not limited to single main or interaction effects. Instead, it describes the overall significance of the global interaction structure, which makes this approach systematic. To identify bladder cancer specific pathway interactions, we adopt the SNP interaction network from this previous study and the network is shown in Figure 1.

### Dataset

The microarray dataset used in this study is publicly available from Gene Expression Omnibus (GEO) website (dataset ID: GDS1479) and more details about this dataset can be obtained at Dyrskjot et al. (2003). Briefly, the original study profile about 22,000 genes to analyze bladder biopsies of superficial transitional cell carcinomas with or without surrounding carcinoma in situ (CIS) lesions and muscle

invasive carcinomas (mTCC). To match disease stages of individuals in the bladder cancer SNP dataset, we only use two out of the five groups in the original study. One group contains 15 tumor biopsies from superficial transitional cell carcinoma (sTCC) without surrounding CIS, which serves as cases. The other group contains 9 biopsies of normal bladder mucosa from patients without a bladder cancer history, which serves as controls. The cRNA from different samples are hybridized to Affymetrix U133A GeneChips. After data processing as described by Dyrskjot et al. (2003), there are about 22,000 genes in the dataset.

### Differential co-expression network

Genes with similar expression patterns may form complexes, pathways, or participate in regulatory and signaling circuits (Eisen et al., 1998; Ideker et al., 2002; Huang et al., 2007). Therefore, gene co-expression networks, which describe the pairwise relations among gene expression profiles, have become a popular tool for microarray analysis. A gene co-expression network is an undirected graph, where the vertices correspond to genes, and edges between genes represent significant co-expression relationships (Stuart et al., 2003). The weight of an edge can be computed using different correlation measures and an edge is included in the network if its weight pass a certain pre-specified threshold.

Gene co-expression networks have been successfully applied in many studies mostly to identify functional gene modules (Stuart et al., 2003; Presson et al., 2008; Weston et al., 2008). However, to study a particular disease, it is the difference between cases and controls that provides the most information about the underlying mechanism. In other words, rather than asking ‘what parts of the system are the most abundant or dominant’, we should ask ‘What parts of the system are most distinctive between different conditions’ (Ideker and Krogan, 2012). Therefore, to gain insights into the transition from healthy individuals to bladder cancer patients, we adapt the framework of the differential network previously proposed (Bandyopadhyay et al., 2010). Instead of constructing two static co-expression networks, one for cases and one for controls, we build differential co-expression networks to describe the changes of pairwise relations among genes from controls to cases. In this way, co-expressions present in both conditions are downplayed or removed from the differential network. The co-expressions that reflect the changes from controls to cases are distinguished from those that support the housekeeping functions.

Specifically, we filter the probes in the microarray dataset and only 308 of them, which also exist in the SNP dataset, are considered in this section. We compute Spearman’s correlation for all  $\binom{308}{2} = 47,278$  pairs of transcripts separately in cases ( $C_{\text{case}}$ ) and controls ( $C_{\text{controls}}$ ). Pairwise differential co-expression is calculated as  $C_{\text{case}} - C_{\text{control}}$  and assigned to the corresponding edge as its weight. The negative

differential co-expression network is constructed to include only edges with significant negative weights and the positive differential co-expression network only include edges with significant positive weights. In other words, the negative differential co-expression network describes the gene pairs which lose correlations from controls to cases, while the positive differential co-expression network describes those which gain correlations.

### Pathway interaction network

We construct pathway interaction network  $G_1$  from SNP interaction network,  $G_2^-$  from negative differential co-expression network and  $G_2^+$  from positive differential co-expression network. In all pathway interaction networks, a vertex represents a pathway. Two pathways are connected if there are more edges than by chance between the two corresponding pathways in the underlying biomarker network.

For example, to construct pathway interaction network  $G_1$ , we use permutation test to determine whether edges are more likely to occur between two pathways than by chance in the SNP interaction network. First, we annotate all the vertices in the SNP interaction network with their functions using canonical pathway annotations from MsigDB collection 2. All SNPs in this dataset are within coding region, so a SNP-to-gene-to-pathway mapping is straight forward. To investigate the enrichment of pathway-pathway interactions, we count the frequency of edges in the SNP interaction network between two particular pathways. Note that pathways with more SNPs might display more edges in the SNP interaction network. To control the bias introduced by pathway sizes, we use permutation test to generate a null distribution. Specifically, we randomly shuffle the SNP-pathway(s) annotations for 1,000 times and assess the frequency of edges between two particular pathways in the original SNP interaction network at each time. The  $P$ -value of a pathway-pathway interaction enrichment is computed as the fraction of corresponding edge frequencies on permuted data which are no smaller than that of the real data. Similarly, pathway interaction networks  $G_2^-$  and  $G_2^+$  are generated based on the negative and positive differential co-expression networks accordingly. In all three cases, a pathway pair is included in the pathway interaction network if the  $P$ -value of the pathway-pathway interaction enrichment is smaller than 0.001.

### Essential vertices identification

To identify key vertices in the networks, we use bottleneck vertices to survey independent regulation as it has been shown that bottleneck vertices are regulated in a condition-dependent manner in biological networks (Yu et al., 2007). We define bottleneck vertices as vertices with the 10 highest betweenness centrality scores. The betweenness centrality score is based on the number of shortest paths that cross a given vertex, and thus reflects how embedded (i.e. central) a vertex is in the network (Freeman, 1977).

Hubs, highly connected vertices in other words, play an integral role in maintaining network integrity and in mass information transfer. Therefore, we also identify hubs of the networks. In this study, we define hub vertices as vertices with the 10 highest degree scores.

Both hub and bottleneck vertices have been associated with functional essentiality (Yu et al., 2007; Zotenko et al., 2008). We are interested in their ability to influence the propagation of signals across the network and their importance to maintain the integrity of the network. Particularly, we define hub-bottleneck vertices as the overlapping vertices between hub vertices and bottleneck vertices.

## Results

### Pathway interaction network derived from SNP interaction network

We study whether SNP-SNP interactions are more likely to occur between certain pathway pairs or not. As described previously, we construct pathway interaction network  $G_1$  by only including pathway pairs which have more edges between them in the SNP interaction network than by chance ( $P < 0.001$ ). Figure 2 shows the structure of pathway interaction network  $G_1$ .  $G_1$  has 386 vertices and 635 edges. There are two connected components in the network and the vertices cluster into several communities. Moreover, there is no self-loop in  $G_1$ , which indicates that SNP-SNP interactions mostly happen between different pathways instead of within the same pathway.

A heavy-tail degree distribution of  $G_1$  indicates the existence of hub vertices. The hub-bottleneck vertices (pathways) are reported in Table 1. Among all the hub vertices, KEGG\_FOCAL\_ADHESION is connected to 269 neighbors and displays the highest betweenness centrality.

Pathway	Betweenness Centrality	Degree
KEGG_FOCAL_ADHESION	0.93	268
BIOCARTA_VITCB_PATHWAY	0.11	32
KEGG_STEROID_HORMONE_BIOSYNTHESIS	0.09	13
REACTOME_BASIGIN_INTERACTIONS	0.07	53
BIOCARTA_INTEGRIN_PATHWAY	0.06	51
BIOCARTA_NO1_PATHWAY	0.06	51
KEGG_VIRAL_MYOCARDITIS	0.05	44
REACTOME_PACKAGING_OF_TELOMERE_ENDS	0.02	10
REACTOME_TELOMERE_MAINTENANCE	0.02	10

Table 1: Hub-bottleneck vertices (pathways) in pathway interaction network  $G_1$  derived from SNP interaction network. Pathways are ranked in descending order of betweenness centrality.

### Differential co-expression network

To study the bladder cancer specific co-expression patterns, we construct differential co-expression networks from the microarray dataset. To match the SNP dataset, only 308 transcripts that also exist in the SNP dataset are used in this section. The changes of Spearman correlation between all

$\binom{308}{2} = 47,278$  pairs of transcripts from controls to cases are shown in Figure 3. The differential co-expression correlation,  $(C_{\text{case}} - C_{\text{control}})$ , follows a normal distribution with  $\text{mean} \pm \text{stdv} = 0.0006 \pm 0.51214$ . The 95% confidence interval is  $[-1.003, 1.004]$ . Transcript pairs with  $C_{\text{case}} - C_{\text{control}} < -1.003$  are included in the negative differential co-expression network, which means that there are co-expressions in controls but not in cases. Similarly, transcript pairs with  $C_{\text{case}} - C_{\text{control}} > 1.004$  are included in the positive differential co-expression network, which indicates that there are co-expressions in cases but not in controls.

The negative differential co-expression network has 297 vertices and 1,114 edges. Among two connected components, the largest one has 293 vertices. Similarly, the positive differential co-expression network possesses 287 vertices and 1,114 edges. There are no isolated islands in the network. Both differential co-expression networks display a heavy-tail degree distribution, which indicates the existence of hub genes.

To identify the key players for maintaining network integrity and signal propagation across the network, we identify the hub-bottleneck vertices in both negative and positive differential co-expression networks (Table 2). Interestingly, gene GATA3, ERBB2 and HFE are identified in both differential co-expression networks.

Gene	Betweenness Centrality	Degree
GATA3	0.21	67
MSH2	0.13	47
ERBB2	0.07	31
HFE	0.04	23
HADHA	0.04	21
LIG1	0.03	26

Gene	Betweenness Centrality	Degree
HFE	0.13	37
FZD7	0.09	26
FOXC1	0.06	53
ERBB2	0.06	27
GATA3	0.06	28

Table 2: Hub-bottleneck vertices (genes) in negative (top) and positive (bottom) differential co-expression networks. Genes are ranked in descending order of betweenness centrality.

### Pathway interaction network derived from differential co-expression network

We investigate whether edges in the differential co-expression networks are more likely to occur between particular pathway pairs than by chance. As described previously, we construct pathway interaction network  $G_2^-$  by only including pathway pairs which have more edges between them in the negative differential network than by chance ( $P < 0.001$ ). Similarly, pathway interaction network  $G_2^+$  is obtained from the positive differential co-expression network.

As shown in Figure 4, pathway interaction network  $G_2^-$  has 78 vertices and 100 edges. There are three connected

components and the entire network divide into several communities. The hub-bottleneck vertices (pathways) are reported in Table 3. Meanwhile, pathway interaction network  $G_2^+$  has 70 vertices and 83 edges. The network has six connected components (Figure 5). Surprisingly, no common edge is shared by  $G_2^-$  and  $G_2^+$ . This means that when comparing cases with controls, the gaining of co-expression and the loss of co-expression happen between different pathway pairs.

Although  $G_2^-$  and  $G_2^+$  do not share any common edge, the essential vertices in the two networks overlap with differences (Table 3). Two pathways, BIOCARTA\_GATA3\_PATHWAY and KEGG\_MELANOGENESIS, are identified as hub-bottleneck vertices in both networks.

It is also an interesting thing that there is no self-loop in  $G_2^-$  (Figure 4) and only two self-loops are observed in  $G_2^+$  (Figure 5). In other words, no pathway possesses more edges within itself than by chance in the negative differential co-expression network. Only two pathways, REACTOME\_GPCR\_LIGAND\_BINDING and SA\_CASPASE\_CASCADE, have more edges within themselves than by chance in the positive differential co-expression network. This indicates that differential co-expressions seem to happen mostly between different pathways rather than within the same pathway.

Pathways	Betweenness Centrality	Degree
KEGG_MISMATCH_REPAIR	0.72	20
BIOCARTA_GATA3_PATHWAY	0.58	24
KEGG_ERBB_SIGNALING_PATHWAY	0.55	12
KEGG_MELANOGENESIS	0.34	13
REACTOME_TRANSCRIPTION	0.19	5
REACTOME_RNA_POLYMERASE_I_III_AND_MITOCHONDRIAL_TRANSCRIPTION	0.19	5
REACTOME_SEMA4D_INDUCED_CELL_MIGRATION_AND_GROWTH_CONE_COLLAPSE	0.15	9

Pathways	Betweenness Centrality	Degree
BIOCARTA_GATA3_PATHWAY	1.0	13
KEGG_MELANOGENESIS	0.90	31
REACTOME_RNA_POLYMERASE_I_PROMOTER_OPENING	0.83	5
REACTOME_CLASS_B2_SECRETIN_FAMILY_RECEPTORS	0.08	8
REACTOME_CLASS_A1_RHODOPSIN_LIKE_RECEPTORS	0.06	5
REACTOME_PEPTIDE_LIGAND_BINDING_RECEPTORS	0.06	5

Table 3: Hub-bottleneck vertices (pathways) in pathway interaction networks  $G_2^-$  (top) and  $G_2^+$  (bottom). The two pathway interaction networks are derived from negative (top) and positive (bottom) differential co-expression networks accordingly. Pathways are ranked in descending order of betweenness centrality.

## Discussion

We have identified pathway-pathway interactions that are associated with bladder cancer risk. Specifically, we investigate pathway interaction at two levels: genetic inter-

actions and co-expression relations. We construct one pathway interaction network from the SNP interaction network and two from the differential co-expression networks. There are some limitations to our approach that are worth highlighting. First, there are multiple other ways pathways can interact with each other, i.e. through sharing common components. Examining pathway relationships more thoroughly at different levels would be helpful. Second, co-expression relationships are transferable. In other words, if two genes are both correlated with a third gene, a correlation will be observed between those two genes. Using partial correlation instead of correlation itself might be helpful to identify direct gene-gene co-expression relationships (de la Fuente et al., 2004).

Despite these limitations, our analyses have helped to identify the pathway interactions that are associated with bladder cancer risk and highlight the importance of pathway interactions. In our study, we observe several interesting aspects and discuss them as follows.

First, we characterize the differential co-expression relationships between cases and controls. The hub-bottleneck genes are identified in the differential co-expression networks. Three genes, GATA3, ERBB2 and HFE, are recognized in both positive and negative differential co-expression networks. Gene GATA3 encodes trans-acting T-cell-specific transcription factor GATA-3, which is an important regulator of T-cell development and plays an important role in endothelial dysplasia. Previously, GATA3 has been suggested as a marker of urothelial differentiation (Higgins et al., 2007; Miyamoto et al., 2012). The fact that GATA3 appears as a hub-bottleneck gene in both positive and negative differential co-expression networks (Table 2) suggests that GATA3 could be actively turning on or off its neighbor genes' expression, which might be highly associated with bladder cancer risk. Whether GATA3 is causing the expression level changes of its neighbor genes and how its neighbor genes contribute to the disease risk should be further studied. Gene ERBB2 encodes a member of the epidermal growth factor (EGF) receptor family of receptor tyrosine kinases. Tumor-specific overexpression of ERBB receptors or their isoforms has been reported in a previous study (Junttila et al., 2003). Whether and how ERBB2 influences its neighbor genes' expressions in the context of bladder cancer risk should be explored. Gene HFE encodes a membrane protein HLA-h, which binds to transferrin receptor (TFR) and reduces its affinity for iron-loaded transferrin (Feder et al., 1998). Whether and how HFE is associated with bladder cancer risk remains unknown. The central role of HFE in the differential co-expression networks might reflect its association with bladder cancer risk.

Second, traditional pathway analyses focus on identifying pathways that are enriched for significant biomarkers. We took a different route and look for pathway pairs that are enriched for between-pathway SNP-SNP interac-

tions or between-pathway differential co-expressions. Consequently, we identify pathways that are "central" in the whole system instead of pathways that are individually associated with the disease risk. We run Gene Set Enrichment Analysis (Subramanian et al., 2005) on both the SNP dataset and the microarray dataset. Most of the pathways reported in Table 1 and 3 are not enriched in the significant biomarkers. However, existing knowledge indicates that some of them could be highly involved in the underlying mechanisms of bladder cancer. For instance, two telomere related pathways are identified as hub-bottleneck vertices in  $G_1$  (Table 1). It is well known that telomere dysfunction or loss can cause sister-chromatid fusions that is associated with oncogene amplification (Campbell et al., 2010; Murnane, 2012).

Third, it is interesting that all networks constructed in this study possess a heavy-tail degree distribution. In other words, most vertices in the networks have only few neighbors whereas a few vertices have many neighbors. This structure makes the network robust to random removal of vertices and vigorous to external perturbations (Barabasi and Bonabeau, 2003; Wang and Chen, 2002). Scale-free structures have been observed in various biological networks (Jeong et al., 2001; Li et al., 2004). It is interesting that we observe a similar structure at the pathway level.

Fourth, we find that the pathway interaction network  $G_1$  does not share any common edge with  $G_2^+$  or  $G_2^-$ . The hub-bottleneck vertices in  $G_1$  are not as "central" in  $G_2^-$  or  $G_2^+$ . Although genetic interactions are very different with co-expression patterns, this result is still surprising. This means that the interactions across pathways are of distinguishing patterns at different levels.

Last, we find that both SNP-SNP interactions and differential co-expressions mostly happen between different pathways rather than within the same pathway. This result suggests that SNP interactions and co-expression patterns in known pathways stay stable across cases and controls. It is the SNP interactions and co-expression patterns between these pathways that are reprogrammed across different disease conditions. Previous studies in yeast have shown that static genetic interactions are enriched within known pathways (Kelley and Ideker, 2005), whereas differential genetic interactions are much more likely to occur among pairs of genes connecting two different pathways than among pairs of genes within the same pathway (Bandyopadhyay et al., 2010). Although co-expressions do not necessarily indicate genetic interactions, it is still encouraging to observe similar results in human SNP and microarray data in our study. Also, this observation further emphasizes the important role of pathway interactions in disease association studies, which highlights the significance of our work.

In summary, we construct bladder cancer specific pathway interaction networks from both SNP-SNP interactions and gene-gene co-expression patterns. Our study highlights key pathway interactions that should be further investigated



and emphasizes the importance of disease-specific pathway interactions.

## Acknowledgements

This work was supported by NIH grants R01-LM009012, R01-LM010098, and R01-AI59694.

## References

- Bandyopadhyay, S., Mehta, M., Kuo, D., and etc. (2010). Rewiring of genetic networks in response to dna damage. *Science*, 330(6009):1385–1389.
- Barabasi, A. L. and Bonabeau, E. (2003). Scale-free networks. *Scientific American*, 288:60–69.
- Boyle, E. I., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J. M., and Sherlock, G. (2004). Go::termfinder—open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics*, 20(18):3710–3715.
- Campbell, P. J., Yachida, S., Mudie, L. J., and etc. (2010). The patterns and dynamics of genomic instability in metastatic pancreatic cancer. *Nature*, 467(7319):1109–1113.
- de la Fuente, A., Bing, N., Hoeschele, I., and Mendes, P. (2004). Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20(18):3565–3574.
- Dyrskjot, L., Thykjaer, T., Kruhoffer, M., Jensen, J. L., Marcussen, N., Hamilton-Dutoit, S., Wolf, H., and Orntoft, T. F. (2003). Identifying distinct classes of bladder carcinoma using microarrays. *Nat Genet*, 33(1):90–96.
- Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A*, 95(25):14863–14868.
- Feder, J. N., Penny, D. M., Irrinki, A., Lee, V. K., Lebron, J. A., Watson, N., Tsuchihashi, Z., Sigal, E., Bjorkman, P. J., and Schatzman, R. C. (1998). The hemochromatosis gene product complexes with the transferrin receptor and lowers its affinity for ligand binding. *Proc Natl Acad Sci U S A*, 95(4):1472–1477.
- Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41.
- Guo, X. and Wang, X.-F. (2009). Signaling cross-talk between tgfbeta/bmp and other pathways. *Cell Res*, 19(1):71–88.
- Higgins, J. P. T., Kaygusuz, G., Wang, L., and etc. (2007). Placental s100 (s100p) and gata3: markers for transitional epithelium and urothelial carcinoma discovered by complementary dna microarray. *Am J Surg Pathol*, 31(5):673–680.
- Hirschhorn, J. N. (2009). Genomewide association studies—illuminating biologic pathways. *N Engl J Med*, 360(17):1699–1701.
- Hu, T., Sinnott-Armstrong, N. A., Kiralis, J. W., Andrew, A. S., Karagas, M. R., and Moore, J. H. (2011). Characterizing genetic interactions in human disease association studies using statistical epistasis networks. *BMC Bioinformatics*, 12:364.
- Huang, Y., Li, H., Hu, H., Yan, X., Waterman, M. S., Huang, H., and Zhou, X. J. (2007). Systematic discovery of functional modules and context-specific functional annotation of human genome. *Bioinformatics*, 23(13):i222–9.
- Ideker, T. and Krogan, N. J. (2012). Differential network biology. *Mol Syst Biol*, 8:565.
- Ideker, T., Ozier, O., Schwikowski, B., and Siegel, A. F. (2002). Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18 Suppl 1:S233–40.
- Ideker, T. and Sharan, R. (2008). Protein networks in disease. *Genome Res*, 18(4):644–652.
- Isserlin, R., Merico, D., Alikhani-Koupaei, R., Gramolini, A., Bader, G. D., and Emili, A. (2010). Pathway analysis of dilated cardiomyopathy using global proteomic profiling and enrichment maps. *Proteomics*, 10(6):1316–1327.
- Jeong, H., Mason, S. P., Barabasi, A. L., and Oltvai, Z. N. (2001). Lethality and centrality in protein networks. *Nature*, 411(6833):41–42.
- Junttila, T. T., Laato, M., Vahlberg, T., and etc. (2003). Identification of patients with transitional cell carcinoma of the bladder overexpressing erbb2, erbb3, or specific erbb4 isoforms: real-time reverse transcription-pcr analysis in estimation of erbb receptor status from cancer patients. *Clin Cancer Res*, 9(14):5346–5357.
- Kelley, R. and Ideker, T. (2005). Systematic interpretation of genetic interactions using protein networks. *Nat Biotechnol*, 23(5):561–566.
- Khatri, P., Sirota, M., and Butte, A. J. (2012). Ten years of pathway analysis: current approaches and outstanding challenges. *PLoS Comput Biol*, 8(2):e1002375.
- Li, S., Armstrong, C. M., Bertin, N., and etc. (2004). A map of the interactome network of the metazoan c. elegans. *Science*, 303(5657):540–543.
- Li, Y., Agarwal, P., and Rajagopalan, D. (2008). A global pathway crosstalk network. *Bioinformatics*, 24(12):1442–1447.
- Lu, L. J., Sboner, A., Huang, Y. J., Lu, H. X., Gianoulis, T. A., Yip, K. Y., Kim, P. M., Montelione, G. T., and Gerstein, M. B. (2007). Comparing classical pathways and modern networks: towards the development of an edge ontology. *Trends Biochem Sci*, 32(7):320–331.
- Miyamoto, H., Izumi, K., Yao, J. L., Li, Y., Yang, Q., McMahon, L. A., Gonzalez-Roibon, N., Hicks, D. G., Tacha, D., and Netto, G. J. (2012). Gata binding protein 3 is down-regulated in bladder cancer yet strong expression is an independent predictor of poor prognosis in invasive tumor. *Hum Pathol*, 43(11):2033–2040.
- Murnane, J. P. (2012). Telomere dysfunction and chromosome instability. *Mutat Res*, 730(1-2):28–36.
- Presson, A. P., Sobel, E. M., Papp, J. C., Suarez, C. J., Whistler, T., Rajeevan, M. S., Vernon, S. D., and Horvath, S. (2008). Integrated weighted gene co-expression network analysis with an application to chronic fatigue syndrome. *BMC Syst Biol*, 2:95.



- Ramanan, V. K., Shen, L., Moore, J. H., and Saykin, A. J. (2012). Pathway analysis of genomic data: concepts, methods, and prospects for future development. *Trends Genet*, 28(7):323–332.
- Schadt, E. E., Friend, S. H., and Shaywitz, D. A. (2009). A network view of disease and compound screening. *Nat Rev Drug Discov*, 8(4):286–295.
- Stuart, J. M., Segal, E., Koller, D., and Kim, S. K. (2003). A gene-coexpression network for global discovery of conserved genetic modules. *Science*, 302(5643):249–255.
- Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*, 102(43):15545–15550.
- Tong, A. H. Y., Lesage, G., Bader, G. D., and etc. (2004). Global mapping of the yeast genetic interaction network. *Science*, 303(5659):808–813.
- Wang, X. F. and Chen, G. (2002). Synchronization in scale-free dynamical networks: robustness and fragility. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 49(1):54–62.
- Weston, D. J., Gunter, L. E., Rogers, A., and Wulschleger, S. D. (2008). Connecting genes, coexpression modules, and molecular signatures to environmental stress phenotypes in plants. *BMC Syst Biol*, 2:16.
- Yu, H., Kim, P. M., Sprecher, E., Trifonov, V., and Gerstein, M. (2007). The importance of bottlenecks in protein networks: correlation with gene essentiality and expression dynamics. *PLoS Comput Biol*, 3(4):e59.
- Zotenko, E., Mestre, J., O’Leary, D. P., and Przytycka, T. M. (2008). Why do hubs in the yeast protein interaction network tend to be essential: reexamining the connection between the network topology and essentiality. *PLoS Comput Biol*, 4(8):e1000140.

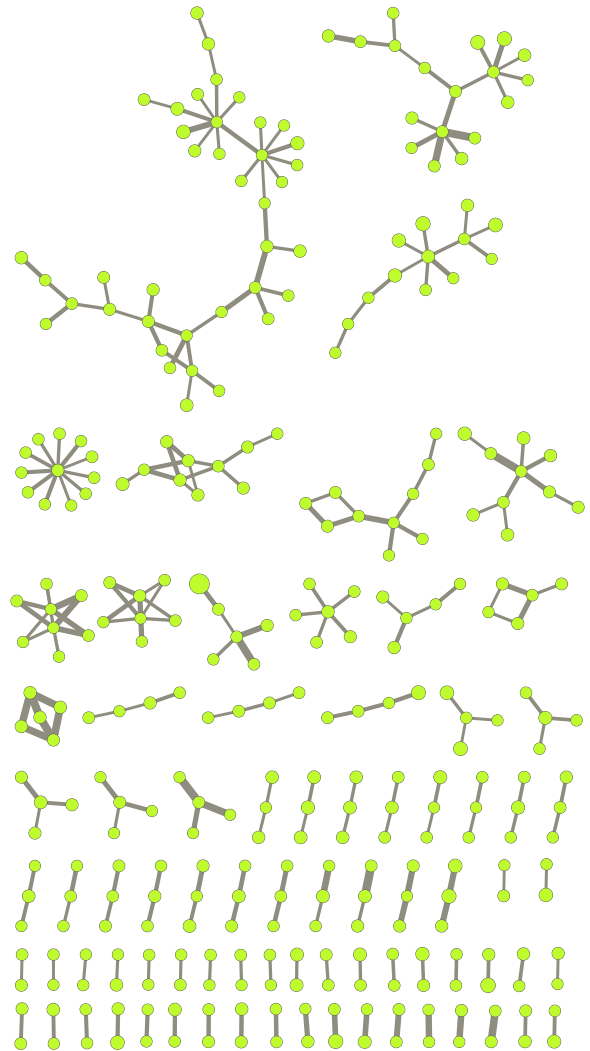


Figure 1: SNP interaction network. A vertex represents a SNP and an edge represents the interaction between the two vertices it connects. The weight of an edge reflects the pairwise interaction strength. There are 319 vertices and 255 edges in the SNP interaction network. The 319 SNPs cover 185 genes. The width of an edge and the size of a vertex are proportional to their weights. More details about this network are available at Hu et al. (2011)

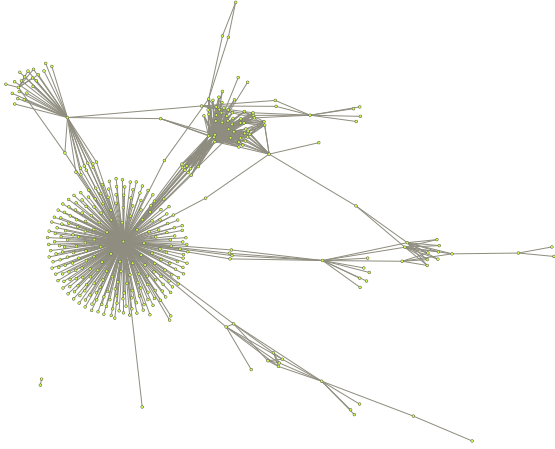


Figure 2: Pathway interaction network  $G_1$  revealed by SNP interaction network. A vertex represents a pathway and an edge exists if the number of SNP interactions between the two corresponding pathways is significantly larger than by chance ( $P < 0.001$ ). There are 386 vertices and 635 edges in the network. The largest connected components has 384 vertices. The network shows a heavy tail degree distribution.

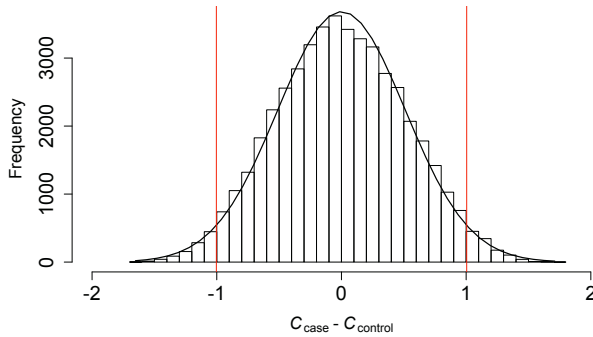


Figure 3: Frequency distribution of pairwise differential co-expression,  $C_{\text{case}} - C_{\text{control}}$ . Spearman correlation for all 47,278 pairs of genes are calculated for cases ( $C_{\text{case}}$ ) and controls ( $C_{\text{control}}$ ) separately and the difference ( $C_{\text{case}} - C_{\text{control}}$ ) are presented.  $C_{\text{case}} - C_{\text{control}}$  ranges from -1.652 to 1.711 and follows a normal distribution. Red lines represent the 95% confidence interval ( $C_{\text{case}} - C_{\text{control}} = -1.003$  and 1.004 respectively). There are 1,011 pairs of transcripts on the left 2.5% tail and 1,114 on the right 2.5% tail.

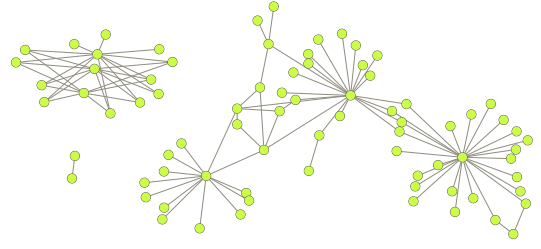


Figure 4: Pathway interaction network  $G_2^-$  revealed by negative differential co-expression network. A vertex represents a pathway and an edge exists if differential co-expression are more likely to occur between the two pathways it connects than by chance in the negative differential co-expression network. The network  $G_2^-$  has 78 vertices and 100 edges.

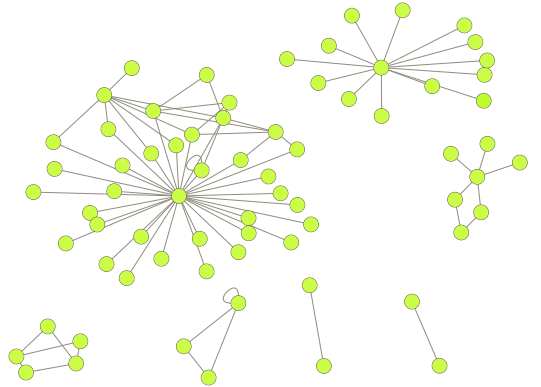


Figure 5: Pathway interaction network  $G_2^+$  revealed by positive differential co-expression network. A vertex represents a pathway. An edge indicates that there are more edges between the two pathways it connects than by chance in the positive differential co-expression network. There are 70 vertices and 83 edges in the network. The vertices fall into six connected components.

# Information Aggregation Mechanisms in Social Odometry

Roman Miletitch<sup>1</sup>, Vito Trianni<sup>3</sup>, Alexandre Campo<sup>2</sup> and Marco Dorigo<sup>1</sup>

<sup>1</sup>IRIDIA, CoDE, Université Libre de Bruxelles, Belgium

<sup>2</sup>Unit of Social Ecology, Université Libre de Bruxelles, Belgium  
{rmiletitch,acampo,mdorigo}@ulb.ac.be

<sup>3</sup>ISTC, National Research Council, Rome, Italy  
vito.trianni@istc.cnr.it

## Abstract

Multi-robot exploration and navigation is a challenging task, especially within the swarm robotics domain, in which the individual robots have limited capabilities and have access to local information only. An interesting approach to exploration and navigation in swarm robotics is *social odometry*, that is, a cooperative strategy in which robots exploit odometry for individual navigation, and share their own position estimation through peer-to-peer local communication to collectively reduce the estimation error. In this paper, the robots have to localize both a home and a goal location and navigate back and forth between them. The way in which navigational information is aggregated influences both the efficiency in navigation between the two areas, and the self-organized selection of better paths. We propose three new parameter-free mechanisms for information aggregation and we provide an extensive study to ascertain their properties in terms of navigation efficiency and collective decision.

## Introduction

Navigation is a basic task for robots in most application domains, would that be for cleaning a room or demining a field. In few cases, the environment is completely known in advance, and therefore a detailed navigation plan can be produced. Most often, the environment is not completely known and exploration is required to identify and reach the desired locations. When multiple robots explore an unknown environment, cooperative strategies can be used to improve exploration and navigation efficiency. This is particularly useful in the swarm robotics domain, in which individual robots cannot rely on global information or complex algorithms (Brambilla et al., 2013). In this paper, we study a cooperative exploration and navigation strategy based on the peer-to-peer exchange of information among robots in a swarm. We propose three variants of the information aggregation mechanism, and we investigate their impact over the dynamics of navigation of the swarm as well as the resulting efficiency with respect to an exploration and exploitation task.

Exploration and navigation strategies in swarm robotics should present a low complexity to match the limited capabilities of the individual robots. The simplest way to explore

and navigate in a closed area is through random walk. While not being the most efficient way, it assures that the robots reach every part of the environment, even if this may require a long time. In order to improve over a purely random exploration, the robots can memorize and map their surroundings to avoid previously explored zones (Thrun, 2008) to reach specific areas of interest. To this purpose, the robot can position itself on the map and navigate in the environment using dead-reckoning techniques such as odometry. Odometry relies on the integration over time of the movement vector—as perceived through the robot (proprioceptive) sensors—in order to maintain an estimate of the robot position. However, this approach is quite error prone since estimation errors are cumulated over time, therefore requiring techniques for error reduction such as Kalman filters (Thrun et al., 2005).

Alternatively, the estimation error can be reduced through the shared effort of multiple robots exchanging structured information (Martinelli et al., 2005). By sharing the estimated position of a landmark, the robots can collectively reduce the overall odometric error. This is a straightforward mechanism that easily lends itself to implementation on very simple robots. Therefore, the collective reduction of odometry errors can be instantiated also in swarm robotics contexts, as it complies with the inherent limitations of the robots. This mechanism was first introduced by Gutiérrez et al. (2009) and is referred to as *social odometry*. In this approach, the robots estimate the navigation path between two target areas in the environment (*i.e.*, home and goal locations) using odometry and attach to this estimate a confidence level that decreases with the distance travelled. At the same time, the robots share their navigation information within the swarm in a local peer-to-peer manner. Thanks to this process, information about target areas spreads gradually within the swarm, contributing to reduce the error in the position estimation. Overall, this decentralized process results in an increased efficiency in the swarm navigation abilities.

An interesting aspect of social odometry is that it naturally leads to the emergence of collective decisions within the swarm (Gutiérrez et al., 2010). Indeed, when there are multiple goal areas to localize (*e.g.*, multiple resources to

exploit), by sharing the available information the robots not only improve the accuracy of their localization but can also decide which area to target. The sum of individual decisions leads to a self-organized behaviour that makes the swarm choose between focusing on a single area/resource or exploiting in parallel several ones.

The efficiency of social odometry as a navigation mechanism and the resulting collective dynamics of decision making depend heavily on the way information is shared and aggregated in the robot swarm. In particular, we found that even small variations in some parameters of the individual behaviour may lead to huge differences in the swarm dynamics. For this reason, in this paper we propose three new parameter-free mechanisms for information aggregation and processing, and study them to ascertain their properties in terms of (i) the efficiency in supporting the swarm navigation and (ii) the ability to produce a collective decision when multiple goal locations are present. The results we obtain allow us to understand the properties of the collective behaviours generated by each information processing mechanism. On the basis of such knowledge and depending on the specific needs of the application at hand, a principled choice of the most appropriate mechanism can be made.

## State of the Art

**Navigation in Swarm Robotics** There are various ways to improve navigation through information-sharing within a swarm. Ducatelle et al. (2011) model a swarm as a communication network that propagates relevant information. Each robot in the swarm maintains a table with navigation information about all known robots, similar to how nodes in a mobile ad hoc network maintain routing tables. Then, the robots propagate the available information and use the table to find the best path to reach a target robot within the swarm. Sperati et al. (2011) also study navigation in a swarm robotics context. In this case, communication is performed through visual signals only and therefore the information exchanged is much less structured. For this reason, they used artificial evolution to synthesize effective navigation strategies.

Several studies in swarm robotics implement navigation and exploration algorithms without sharing structured information, sometimes exploiting robots as physical landmarks. Rekleitis et al. (2001) divided the swarm in two teams, one moving and the other stationary, serving as a reference for navigation. The teams alternate between stationary and moving states. Nouyan et al. (2009) exploit robots to form complex structures such as chains, in which one end of the chain connects to a central place while the other end explores the environment. Once the goal location is reached, the chain can be exploited by other robots for navigation purposes, or a bucket brigade method can be used for transporting objects along the chain (Ostergaard et al., 2001).

**Collective Decisions** When there are several goal/resource locations present in the environment, the robots may make a collective decision and focus on the exploitation of a single one. This can be beneficial if it is necessary to aggregate a sufficient number of robots in support of the collective localization, or if exploitation requires several robots at the resource. However, this may lead to congestion (*i.e.*, the path to the resource is overused and robots have trouble navigating) or overexploitation of the resource. In this case, the swarm is better off exploiting several resources in parallel.

In order to agree on one option, the robots can either switch to the best option available in their neighbourhood, or average out all the available information. Social odometry allows doing both simply by tuning a single parameter (Gutiérrez et al., 2010). Olfati-Saber et al. (2007) study the swarm as a multi-agent network and present a theoretical framework for the analysis of consensus algorithms. It is possible to obtain collective decisions also through the amplification of the various opinions present in the swarm. Following this approach, the more an opinion is represented in the swarm, the higher the probability of robots switching their opinion (Garnier et al., 2007, 2009; Montes de Oca et al., 2011). This approach requires gathering the opinion of several neighbours, while social odometry works with peer-to-peer interactions, which is easier to implement.

## Social Odometry & Information Processing

In our experiments, the goal of the robots is to locate both a home area and a goal area and then to efficiently navigate back and forth between them. Once one of these two target areas is discovered, its position is kept in memory and updated using odometry. The information about target areas is shared with other robots upon encounter, following the social odometry mechanism. Within this framework, we study the navigation process, the social dynamics, and the link between the two. In the following, we first describe how robot use the available information (either from individual or social odometry) for navigation purposes. Then, we introduce the information processing mechanisms we have devised.

## The Controller

The behaviour of the robot is defined by a finite state automaton with five states: *Explore*, *Go Home*, *Go to Goal*, *Leave Home*, *Leave Goal* (Fig. 1). Robots start in the *Explore* state and return to it whenever they lack relevant information. The other four states form a loop that corresponds to the robot navigating back and forth between the target areas: go to a target area, enter and leave it, then go to the next one. On top of these control states, both short and long range collision avoidance is implemented.

The robots start without any prior knowledge about the location of the target areas. Therefore, they first have to explore the arena. When in the *Explore* state, the robots perform a random walk until they discover the position of

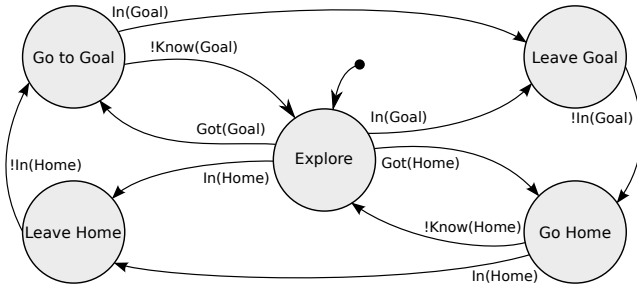


Figure 1: Robot's finite-state automaton. The circles define the states while the arrows define the transitions.  $In(Area)$ ,  $Area \in \{Home, Goal\}$ , is true when a robot senses the grey level of the area,  $Know(Area)$  is true when the robot knows the position of the area,  $Got(Area)$  is true when it just gets this estimation. The robots start in the *Explore* state.

both target areas (home and goal). This can happen in two ways: either they receive relevant information from teammates or they stumble upon a target location ( $Got(Area)$  becomes true, with  $Area \in \{Home, Goal\}$ ). In both the *Go to Goal* and *Go Home* states, the robots move straight to the target location, possibly avoiding other robots and obstacles. Along their travel, they update the target areas location using odometry and update their confidence in the information. The confidence is defined as the inverse of the distance that the robot had travelled from the target area. Therefore, a straight path results in a higher confidence than a curved one.

Once a robot reaches an area (*i.e.*,  $In(Area)$  is true), it traverses it in a straight line (possibly dodging other robots to avoid collisions) and stores the area location. In order to get an estimated position closer to the center of the area, the robot averages its entering and its exiting positions. No matter how many goals there are in the arena, the robots always memorize only one home and one goal (the last seen or agreed upon).

### Information Sharing and Processing

While robots navigate between target areas, they share the information they have on relative locations in order to counterbalance the decrease in information confidence. Not all information is shared at the same time. When in the *Explore* state the robots share the sole information they have. In the other states, the robots share only the information of the last visited location. The information received by a robot is aggregated with the robot's own information. The way this aggregation is performed depends on the information processing mechanism implemented. Given that robots do not share the same reference frame, a transformation is needed. This is made possible by knowing the relative position (range and bearing) of the robot that is sharing its information (for more details, see Gutiérrez et al., 2009). Once the location is obtained in a shared reference frame, the information aggrega-

tion process takes place. Here, we first describe the information aggregation mechanism used by Gutiérrez et al. (2009), and then we introduce our contributed mechanisms.

Let  $i$  and  $j$  be two robots,  $i$  receiving a message from  $j$ . Let  $\mathbf{p}_i$ ,  $\mathbf{p}_j$  be their estimated position of an area (either home or goal) and  $c_i$ ,  $c_j$  the confidence over their respective estimation. The result of any aggregation is the updated couple  $\langle \mathbf{p}_i, c_i \rangle$ . The aggregation mechanism used by Gutiérrez et al. (2009) is based on a Fermi distribution. A weight is calculated from the difference in confidence in order to make a linear combination of the positions:

$$\langle \mathbf{p}_i, c_i \rangle \leftarrow k \cdot \langle \mathbf{p}_i, c_i \rangle + (1 - k) \cdot \langle \mathbf{p}_j, c_j \rangle$$

$$k = \frac{1}{1 + e^{-\beta(c_i - c_j)}}$$

The parameter  $\beta$  measures the importance of the relative confidence levels in the information aggregation. For low values, the aggregation is close to an average, ignoring the confidence. For higher values, the aggregation is stiff: only the information with highest confidence is kept. Finding the right value of  $\beta$  is often a process of trial and error. Our contribution in this paper is the introduction of three parameter-free aggregation mechanisms: *Hard Switch (HS)*, *Random Switch (RS)* and *Weighted Average (WA)*.

**Hard Switch (HS)** In this winner-take-all mechanism, the robots keep the information with highest confidence (either the current information or the received one) and discard the other one. This mimics the Fermi mechanism with a high  $\beta$ .

$$\langle \mathbf{p}_i, c_i \rangle \leftarrow \langle \mathbf{p}_x, c_x \rangle, \quad x = \arg \max_{k \in \{i, j\}} c_k$$

**Random Switch (RS)** As in the mechanism above, here the robots keep one piece of information and discard the other. In this case, however, the switch is stochastic: the higher the confidence, the higher the probability of accepting the information. In practice, this mechanism is a stochastic version of the *HS*.

$$P(\langle \mathbf{p}_i, c_i \rangle \leftarrow \langle \mathbf{p}_j, c_j \rangle) = \frac{c_j}{c_i + c_j}$$

**Weighted Average (WA)** This mechanism consists in a linear combination of both estimated positions with their confidence as weight. On the one hand this implies no loss of information, on the other hand, when information about different goals is aggregated, the new position may not coincide with a real goal location, leading to the apparition of artefacts. While the Fermi mechanism focuses on the difference between the two confidences, here we directly use each of them as weights.

$$\langle \mathbf{p}_i, c_i \rangle \leftarrow \left\langle \frac{c_i \cdot \mathbf{p}_i + c_j \cdot \mathbf{p}_j}{c_i + c_j}, \frac{c_i + c_j}{2} \right\rangle$$



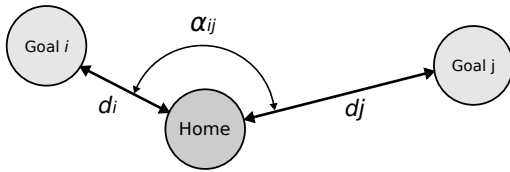


Figure 2: Setup of the experimental arena. The home area is placed in the center of a circular arena of 11 m radius surrounded by walls (not displayed). The goals are characterised by their distance to the home  $d_i, d_j$  and the angles they form with each other  $\alpha_{ij}$ .

## Experiments

We used an experimental setup with as few variables as possible: a circular arena (radius: 11 m) with the home in the center and the goals scattered around (Fig. 2, surrounding walls not shown). The goals are defined by their distance to the home ( $d_i$ ) and the angle between each other ( $\alpha_{ij} \in [\pi/3, \pi]$ ). Both goal and home are of radius 50 cm, and are differently coloured in grey levels to be distinguished by the robots.

Our experiments are performed in the ARGoS open source multi-robot simulator (Pinciroli et al., 2012). The robots we use are the marXbots (Bonani et al., 2010). To accomplish their task, the robots are equipped with several sensorimotor and communication devices. In our experiments, the robots use the infrared ground sensors to check whether they entered an area and to detect its type (home or goal) depending on the area’s grey level. They also use the infrared proximity sensors for short range collision avoidance and the range&bearing device for both communication and long range collision avoidance among robots (Bonani et al., 2010). This last device gives both angle and distance between neighbouring robots and allows them to send short messages. Wheels encoders provide the movement vector for odometry. A simulated gaussian noise with 5% standard deviation models the odometry estimation error. The control loop is executed 10 times per second. Unless stated otherwise, we used 75 robots spawned randomly.

By varying the number of goals, we study different aspects of the collective behaviour, such as the impact of the density of robots on their navigation abilities, the collective decision made by the swarm in a two goals setup, and how this generalizes in multiple goals setups. In the following, we briefly describe the experiments we present in this paper.

**Single Goal** When a single goal is present, we expect that all robots will converge on the same path. The more robots in the arena, the harder it is for them to avoid each other. As density rises, the robots have to handle more and more congestion on their path, which leads them to travel bigger distances and to accumulate more error. This also corresponds to less round trips between home and goal, hence lowering

the efficiency of the swarm. We define the density on a path as the number of robots on it divided by its length.

In order to study the impact of density on navigation, we devised an experimental setup in which we vary both the distance between the home and the goal and the number of robots. All three information processing mechanisms are tested and compared with a benchmark condition in which the robots are provided with perfect information (*PI*) about the goal and home locations. In each experiment, we measure the navigation speed, computed as the number of round trips over time and we study its evolution for values of density between 2 and 40 robots/m. For each density value, we run 100 trials in which we randomly draw the distance between home and goal in the interval [3,8] m, and we compute the corresponding number of robots to obtain the specified density value (which will be in the range [6,320]).

**Two Goals** When there is more than one goal, a decision has to be made as how to spread the robots among the available paths. In this setup, we study if and how the robots converge on one path as well as the implications of such convergence over efficiency. In order to study this decision making process, we count the number of robots committed to each goal, as well as the uncommitted ones. Given that robots do not distinguish between different goals and only store one estimated position  $\mathbf{p}_g$ , a robot is considered to be committed to a goal  $i$  among  $n$  possible if it has information about both goal ( $c_g \neq 0$ ) and home ( $c_h \neq 0$ ), and if goal  $i$  is the closest one to the robot’s estimated goal position  $\mathbf{p}_g$ .

In this setup, we have two goals which can be either at short distance (5 m) or at long distance (8 m). We run experiments with both equal and different distances for the goals: Short/Short (*SS*), Short/Long (*SL*) and Long/Long (*LL*). For each condition, we perform 1000 replications by randomly varying the angle between the sources with  $\alpha_{ij} \in [\pi/3, \pi]$  (cf. Fig. 2).

**Multiple Goals** The environment in which a swarm evolves is rarely as simple as in the two goals setup. Through a multiple goals setup, we enquire about the scalability of the results previously gathered.  $M$  goals are uniformly distributed around the home location, with an angular separation between adjacent goals of  $\pi/M$ , where  $M \in [3, 6]$ . To investigate both the navigation and the decision making abilities, we test three different conditions. Either all goals are at the same distance, short (*SSS*) or long (*LLL*), or a single goal is closer to home (*SLL*). For each condition, we performed 250 trials.

## Results

Each trial in all the previous setups lasts 20 minutes of simulated time. We use the same random initialization in all the runs for the different opinion processing. For each run

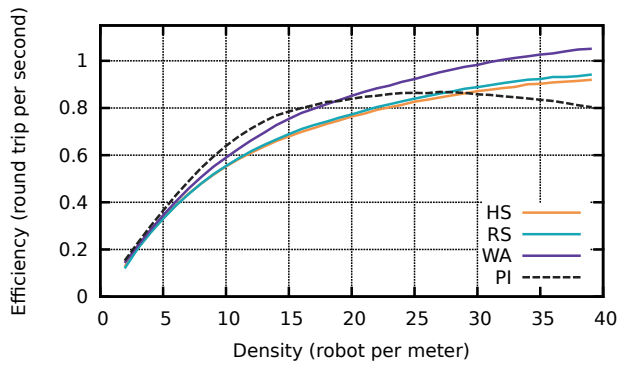


Figure 3: Impact of density on navigation efficiency for each mechanism and in the perfect information control condition. Each line is the mean over 100 trials.

we compute the number of robots on each path to study the dynamics of collective decisions, and the number of round trips to study the navigation efficiency.

### Navigation

As we can see in Fig. 3, all the proposed mechanisms and the control condition with perfect information (*PI*) follow the same tendency. For low densities, we can observe a linear increase in the number of round trips. With higher densities, the growth slows down. As expected, robots with perfect information are the most efficient at first, but their efficiency reaches a peak because of the artefacts created by perfect information. With *PI*, since all robots aim at the center of the target areas (either home or goal), as the density rises they have increased difficulties in avoiding collisions and in entering or exiting the target areas.

Congestion has a lower impact on navigation efficiency with social odometry. In this case, *WA* proves to be more resilient to congestion than *HS* and *RS*. This is due to a smoother navigation in the surrounding of the home and goals, where robots try to enter small and densely populated area. First, since the *WA* mechanism never discards information but averages it, the precision on the estimated position is better than with *HS* or *RS*. Second, the reception of even a slightly better information is smoothly integrated in the *WA* mechanisms, while in both *HS* and *RS* it may cause a large leap of the new location, which may be difficult to reach in case of high densities.

### Collective Decision

Congestion explains why sometimes it is better to spread along multiple paths when there is more than one goal/resource. This decisions impacts not only the efficiency but also the spatial arrangement of the swarm and the way it reacts to changes in the environment.

**Decision** The decision pattern of the swarm results from the sum of local decisions made by the robots. The dynam-

ics of the collective decision are shown in Fig. 4, which plots the convergence pattern generated by the *HS* mechanism when confronted with the *SL* experimental condition. Here the swarm decides to focus on the closest area/resource and most robots converge quickly on the associated path. This behaviour is typical of all three social mechanisms when there is a goal closer to home. We can observe three different phases. At first (0-120 s), most robots are uncommitted and explore for goal areas, reinforcing each as they discover them. Then (120-400 s), a competition among the two alternative paths occurs. The shorter path is reinforced more because of the better information the robots have when encountering robots coming from the other goal. Eventually, the swarm enters a maximization state in which mostly one path is exploited while uncommitted robots continue to join.

Fig. 5 left shows the percentage of robots that choose path A (*i.e.*, the shortest path in the *SL* condition). We note that in the *SL* case, all information aggregation mechanisms lead to convergence on a single path with at least 90% of the robots. Both *HS* and *RS* always lead to a convergence on the closest goal. Similarly for *WA*, which however presents also a low probability to make the robots converge on the distant goal. This happens because with *WA* no information is discarded. When a large number of robots discovers the distant goal early in the experiment, they may influence the whole swarm despite the lower confidence in their information. This cannot happen in the *HS* and *RS*, because low quality information is instantly discarded. In both the *SS* and *LL* experimental conditions, when there is no better choice, *HS* and *RS* lead to a split in the swarm, and robots spread among the two paths (Fig. 5 left). In these experimental conditions, the more robots on a path, the higher the congestion, and the larger the distance the robots travel. This causes robots to have worse confidence in their information with respect to those from a less congested path. Therefore, switches to the other path are very likely. Congestion cre-

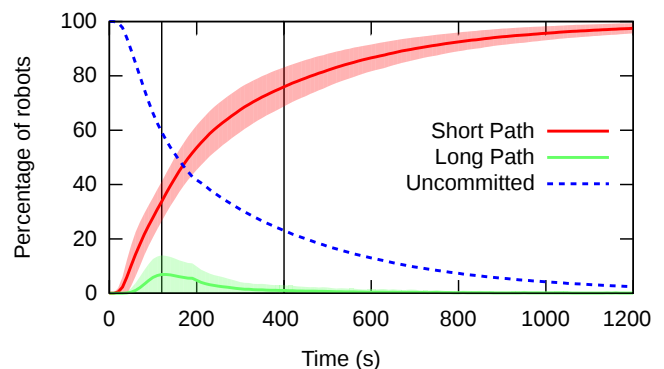


Figure 4: Evolution of the robots repartition between the two target areas using *Hard Switch* in the Short/Long condition. Bold lines indicate the mean over 1000 repetitions, and the shaded areas indicate the standard deviation.

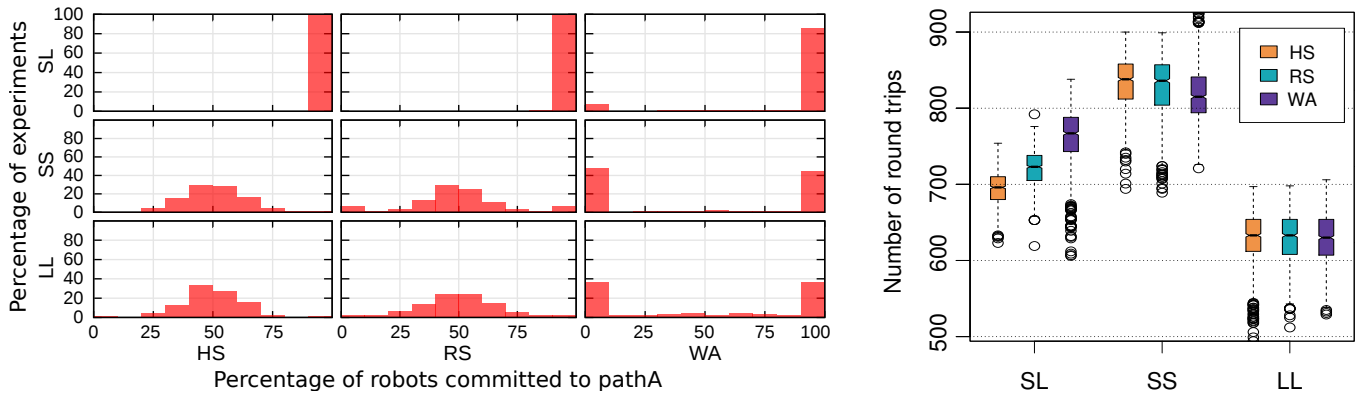


Figure 5: Left: robots repartition on path A. Each histogram shows the observed frequencies of the number of robots committed to path A (the shortest possible path). Right: efficiency of the swarm for two goals, for all mechanisms and conditions. Each box represents the inter-quartile range, whiskers extend to 1.5 times the corresponding quartiles, and dots represent outliers.

ates a sort of negative feedback that leads to an oscillating dynamics in which no decision ends up being taken. On the contrary, WA is not affected by such negative feedback and systematically leads to convergence (randomly on either path, the setup being symmetrical). Indeed, the worse confidence resulting from congestion is counterbalanced by the larger number of robots with which the information is shared and averaged. Therefore, the swarm converges to the more populated path.

**Efficiency** The robot behaviour does not explicitly encode the ability to make collective decisions. Instead, it is conceived to provide efficient navigation ability thanks to the information shared within the swarm. The decision process is an emergent result of this behaviour and so is the variation in efficiency depending on the setup and the mechanisms involved, as shown in Fig. 5 right. In the SL condition, all three mechanisms make the robots converge on the closest path, therefore resulting in density of 15 robots/m. As shown in Fig. 3, WA is more resilient to congestion, and this is why it is the most efficient mechanism in this setup, followed by RS and HS. In the SS condition, both HS and RS result in the swarm splitting between the two paths as discussed above. By exploiting two paths with a low density of 7.5 robots/m (instead of one with high density of 15 robots/m) the robots create less congestion, which explains why the performance for HS and RS is slightly better than in the WA case. Indeed, WA makes the swarm converge on a single path with high density, and navigation is slightly less efficient. Congestion has a lower impact in the LL conditions as both densities (9.4 robots/m on a single path, 4.7 robots/m on two paths) fall in the linear part of the congestion curve (see Fig. 3), explaining why the mechanisms result in the same efficiency.

### Generalization to Multiple Goals

The dynamics we observe with multiple goal locations are similar to the ones displayed in the two goals setup, no mat-

ter the number of added goals. Fig. 6 shows the percentage of robots that choose path A (*i.e.*, the shortest path in the SLL condition), when multiple goal locations are present. All mechanisms leads to convergence in the SLL case, even if WA sometimes leads to the selection of one of the distant goals, for the same reasons discussed in the two goals setup. We can observe a similar splitting behaviour in the SSS and LLL conditions for both HS and RS, while convergence is observed for WA. When the swarm splits, the repartition of robots is not anymore centred on 50% but closer to 33%, meaning that the repartition is not anymore among only two paths. Nonetheless, not all are exploited at the same time, as can be inferred from the existence of paths selected by no robot. This can be explained by the oscillation dynamics discussed earlier. When the amplitude of the oscillations is greater than the number of robots on a path, all the robots on this path switch to another one. This happens in the case of multiple goals because robots are spread among more paths and therefore their number on each is lower.

To better understand the exploitation of the available resources/goals, in Tab. 1 we report the average percentage of robots on the different paths, ordered from the most to the least exploited path. We note that the number of exploited goal locations is most of the time no more than 3. This explains why the efficiency of the swarm does not vary with the number of available resources, as shown in Fig. 7. The slight increase in performance can be attributed to the fact that the more goals there are, the easier it is for uncommitted robots to join a path earlier in the experiment. Overall, we note similar patterns over efficiency between the multiple goals condition and the two goals condition.

When there are multiple goals, WA in the SLL condition leads to a frequent selection of a distant goal instead of the closest one, as shown in Fig. 6. If several distant locations are present, they end up reinforcing each other as their angular distance becomes smaller. In other words, two distant goal locations that are close to each other attract more robots

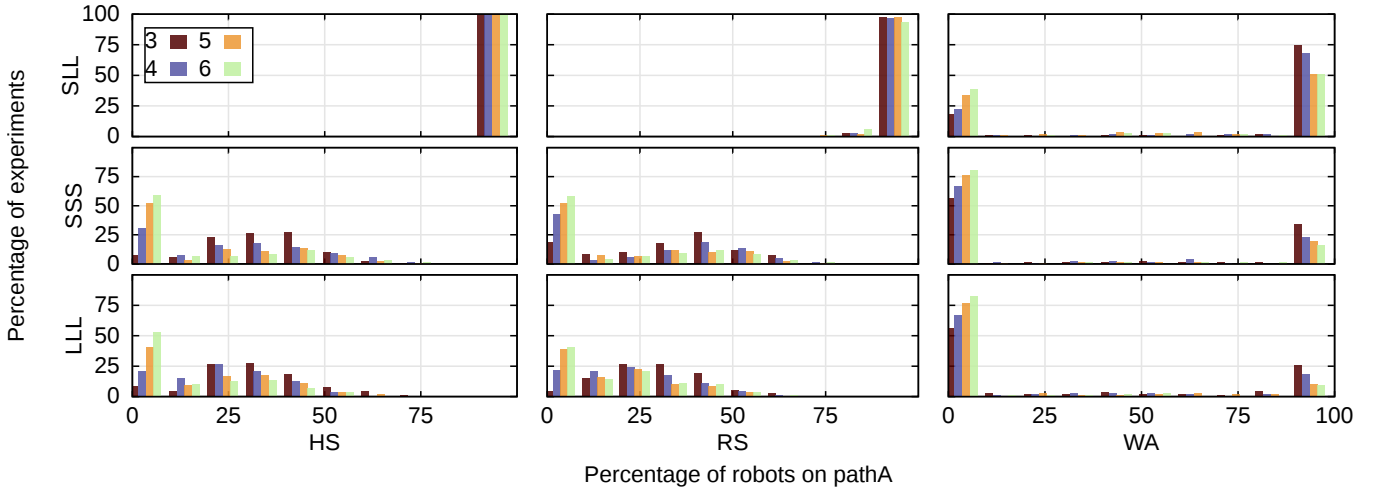


Figure 6: Robots repartition on path A for different number of goal areas (3,4, 5 and 6). Each histogram shows the observed frequencies of the number of robots committed to path A (the shortest possible path).

Table 1: Repartition in percentage of robots for 3, 4, 5 and 6 goals. The 1<sup>st</sup> goal is the one associated with the highest number of robots. The mean and maximum of the standard deviation is (4.7, 10.5) for *HS* and *RS* and (6.1, 16.9) for *WA*.

	<i>SL</i>			<i>SS</i>			<i>LL</i>		
	<i>HS</i>	<i>RS</i>	<i>WA</i>	<i>HS</i>	<i>RS</i>	<i>WA</i>	<i>HS</i>	<i>RS</i>	<i>WA</i>
1 <sup>st</sup>	98.5	98.1	96.0	48.0	52.6	93.0	48.8	47.0	90.8
2 <sup>nd</sup>	0.1	0.6	2.3	34.3	37.3	5.7	33.4	32.5	7.0
3 <sup>rd</sup>	0.0	0.0	0.0	17.2	9.7	0.0	17.4	19.7	0.1
1 <sup>st</sup>	98.4	97.7	95.2	50.6	54.1	92.3	44.8	43.8	89.5
2 <sup>nd</sup>	0.2	1.0	3.6	35.2	38.0	6.8	32.0	31.0	9.2
3 <sup>rd</sup>	0.0	0.1	0.0	12.5	6.9	0.0	17.6	17.2	0.1
4 <sup>th</sup>	0.0	0.0	0.0	2.1	0.7	0.0	5.1	7.0	0.0
1 <sup>st</sup>	98.6	97.3	92.4	51.1	51.1	94.8	44.9	42.6	89.4
2 <sup>nd</sup>	0.2	1.0	6.8	35.2	37.0	4.5	31.6	30.0	9.5
3 <sup>rd</sup>	0.0	0.1	0.0	12.5	10.4	0.2	17.7	17.7	0.5
4 <sup>th</sup>	0.0	0.0	0.0	1.1	1.1	0.0	5.1	7.3	0.0
5 <sup>th</sup>	0.0	0.0	0.0	0.0	0.0	0.0	0.3	1.4	0.0
1 <sup>st</sup>	98.6	97.3	93.6	50.1	53.0	94.7	43.7	42.4	88.5
2 <sup>nd</sup>	0.2	1.5	5.4	34.9	36.1	4.7	31.7	28.2	10.4
3 <sup>rd</sup>	0.0	0.1	0.5	13.5	9.2	0.2	17.2	17.3	0.6
4 <sup>th</sup>	0.0	0.0	0.0	1.4	1.3	0.0	6.0	8.3	0.0
5 <sup>th</sup>	0.0	0.0	0.0	0.1	0.2	0.0	0.8	2.3	0.0
6 <sup>th</sup>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0

than a single closer location. This explains why the chance of *WA* leading to the selection of a distant goal increases with the number of goals.

## Discussion

The experiments above reveal the specificities of the three information aggregation mechanisms. *WA* leads to convergence to a single path in all conditions, but this is slower and error-prone. In the whole, *WA* leads to better cohesion of the

swarm and deals better with congestion thanks to more accurate information about the target areas. *HS* and *RS* also lead to convergence when there is a shorter path to exploit, and handle better the presence of multiple distant goal locations. When congestion results in inefficient navigation, both mechanisms lead to the exploitation of multiple paths, spreading the load of robots in a balanced way with similar dynamics, although *HS* appears to be stiffer than *RS*.

## Conclusions

In this paper, we presented an extensive analysis of three parameter-free information processing mechanisms for social odometry. We studied the impact of these mechanisms on the navigation efficiency and on the dynamics of the swarm. In particular, we observed how the information processing mechanism can either lead to convergence on the exploitation of a single path, or to splitting over multiple comparable options. These results are meant to give future designer a guideline of which mechanism to choose depending on the situation at hand.

In future work, we plan to further investigate the dynamics of social odometry in order to provide an optimal load-balancing behaviour. This would maximize the exploitation of different resources and provide the swarm the ability to react to changes in its environment in real time. Additionally, we will experiment with more complex paths, for instance in the presence of obstacles. Also, physical objects to be retrieved may be placed within the goal areas in order to simulate a more realistic environment and making it possible to test the collective behaviour with real robots. Last, heterogeneity can be added in the swarm. On the one hand, individual robots may get committed to a goal with different individual preferences, leading to a better exploration of the environment. On the other hand, different groups of robots



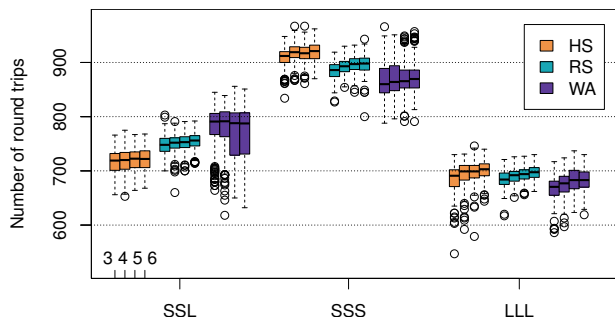


Figure 7: Efficiency of the swarm for multiple goals and all mechanisms and conditions. See Fig. 5 for more details.

could compete for the best source, each of them having different information aggregation mechanisms, leading to a different exploitation of resources among different groups.

### Acknowledgments

This work has been supported by the Italian CNR and the Belgian F.R.S.-FNRS within the EUROCORES Programme EuroBioSAS of the European Science Foundation.

### References

- Bonani, M., Longchamp, V., Magnenat, S., Rétornaz, P., Burnier, D., Roulet, G., Vaussard, F., Bleuler, H., and Mondada, F. (2010). The MarXbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4187–4193. IEEE Press, Piscataway, NJ.
- Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41.
- Ducatelle, F., Di Caro, G., Pinciroli, C., Mondada, F., and Gambardella, L. (2011). Communication assisted navigation in robotic swarms: Self-organization and cooperation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4981–4988. IEEE Computer Society Press, Los Alamitos, CA.
- Garnier, S., Gautrais, J., Asadpour, M., Jost, C., and Theraulaz, G. (2009). Self-organized aggregation triggers collective decision making in a group of cockroach-like robots. *Adaptive Behavior*, 17(2):109–133.
- Garnier, S., Tâche, F., Combe, M., Grimal, A., and Theraulaz, G. (2007). Alice in pheromone land: An experimental setup for the study of ant-like robots. In *Swarm Intelligence Symposium, 2007*, pages 37–44. IEEE Press, New York City, NY.
- Gutiérrez, A., Campo, A., Monasterio-Huelin, F., Magdalena, L., and Dorigo, M. (2010). Collective decision-making based on social odometry. *Neural computing & applications*, 19(6):807–823.
- Gutiérrez, A., Campo, A., Santos, F., Monasterio-Huelin Maciá, F., and Dorigo, M. (2009). Social odometry: imitation based odometry in collective robotics. *International Journal of Advanced Robotic Systems*, 6(2):129–136.
- Martinelli, A., Pont, F., and Siegwart, R. (2005). Multi-robot localization using relative observations. In *IEEE international conference on robotics and automation (ICRA)*, pages 2797–2802. IEEE Press, New York, NY.
- Montes de Oca, M. A., Ferrante, E., Scheidler, A., Pinciroli, C., Birattari, M., and Dorigo, M. (2011). Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making. *Swarm Intelligence*, 5(3-4):305–327.
- Nouyan, S., Groß, R., Bonani, M., Mondada, F., and Dorigo, M. (2009). Teamwork in self-organized robot colonies. *IEEE Transactions on Evolutionary Computation*, 13(4):695–711.
- Olfati-Saber, R., Fax, J. A., and Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233.
- Ostergaard, E. H., Sukhatme, G. S., and Matari, M. J. (2001). Emergent bucket brigading: a simple mechanisms for improving performance in multi-robot constrained-space foraging tasks. In *Proceedings of the fifth international conference on Autonomous agents, AGENTS '01*, pages 29–30, New York, NY. ACM.
- Pinciroli, C., Trianni, V., O’Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Birattari, M., Gambardella, L. M., and Dorigo, M. (2012). ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295.
- Rekleitis, I., Dudek, G., and Milios, E. (2001). Multi-robot collaboration for robust exploration. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):7–40.
- Sperati, V., Trianni, V., and Nolfi, S. (2011). Self-organised path formation in a swarm of robots. *Swarm Intelligence*, 5(2):97–119.
- Thrun, S. (2008). Simultaneous localization and mapping. Springer Tracts in Advanced Robotics, pages 13–41–41. Springer Berlin Heidelberg.
- Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press, Cambridge, MA.



# Rapid Rule Compaction Strategies for Global Knowledge Discovery in a Supervised Learning Classifier System

Jie Tan, Jason H. Moore, Ryan J. Urbanowicz

Geisel School of Medicine, Dartmouth College, 1 Medical Center Dr., Lebanon, NH 03756

Ryan.J.Urbanowicz@dartmouth.edu

## Abstract

Michigan-style learning classifier systems have availed themselves as a promising modeling and data mining strategy for bioinformaticists seeking to connect predictive variables with disease phenotypes. The resulting ‘model’ learned by these algorithms is comprised of an entire population of rules, some of which will inevitably be redundant or poor predictors. Rule compaction is a post-processing strategy for consolidating this rule population with the goal of improving interpretation and knowledge discovery. However, existing rule compaction strategies tend to reduce overall rule population performance along with population size, especially in the context of noisy problem domains such as bioinformatics. In the present study we introduce and evaluate two new rule compaction strategies (QRC, PDRC) and a simple rule filtering method (QRF), and compare them to three existing methodologies. These new strategies are tuned to fit with a global approach to knowledge discovery in which less emphasis is placed on minimizing rule population size (to facilitate manual rule inspection) and more is placed on preserving performance. This work identified the strengths and weaknesses of each approach, suggesting PDRC to be the most balanced approach trading a minimal loss in testing accuracy for significant gains or consistency in all other performance statistics.

## Introduction

Learning classifier systems (LCSs) are an adaptive rule-based class of algorithms which combine evolutionary computing with machine learning and other heuristics (Holland, 1986; Wilson, 1995). More recently, Michigan-style LCSs (M-LCSs) have been shown to be an effective approach for the detection and characterization of complex patterns of association in epidemiological data mining. This work applied M-LCSs to identify patterns of multi-locus interaction (i.e. epistasis) as well as heterogeneity when seeking to connect predictive genetic and environmental variables with human disease phenotypes (Urbanowicz and Moore, 2010; Urbanowicz et al., 2012b, 2013). LCSs yield a resulting model/solution, comprised of an entire population of rules, affording them the ability to learn iteratively and distribute learned patterns across this population. These characteristics make the application of LCSs to the problem of heterogeneous patterns particularly appealing.

A notable effect of this iterative, “one instance at a time”, LCS learning is the transitional nature of the rule population, i.e. the rule population is constantly changing with offspring rules continually being added and rules of lesser fitness being eliminated. At any given learning iteration, an unknown number of rules are bound to exist in the rule population that make little or no contribution to the overall performance of the system. These include (1) rules that overlap in describing the problem space, (2) poor, recently generated rules, that the algorithm has yet to identify as poor (i.e. low accuracy), and (3) conflicting rules that harm overall performance. Additionally, a solution comprised of a population of many rules can make interpretation and knowledge discovery a considerable challenge. Interpretation of an LCS rule population had been traditionally approached with manual rule inspection, i.e. an expert would examine the best rules of a population in an attempt to extract knowledge. With this task in mind, a *rule compaction* strategy that could consolidate the rule population to a minimum set of critical, human readable rules was considered to be useful.

Wilson implemented the first LCS rule compaction strategy applied to his XCSI algorithm. This Compact Rule-set Algorithm (CRA) achieved a much smaller ruleset, yet maintained high training and testing performance when applied to Wisconsin Breast Cancer dataset (Wilson, 2002). Like most strategies that would follow, rule compaction was run following completion of the LCS algorithm as a form of post processing. Wilson’s approach was designed for classifiers that had been highly trained such that the rules were maximally general and always correct in their classification of the test data. CRA was implemented purely to facilitate manual rule inspection by dramatically reducing the rule population size. Later, Fu and Davis revised CRA in order to handle less well trained, noisy classifier systems (Fu and Davis, 2002). The consideration of noisy problems wherein training and testing accuracies might never approach 100% accuracy is critical in problem domains such as bioinformatics and epidemiology. Without doing so, rule compaction is likely to sacrifice performance of the rule population in exchange for minimal size. However, the shortcomings of

CRA and Fu's approaches lie in heavy computation and time complexity, because overall rule population performance needs to be calculated each time a classifier is considered for addition or removal. In an effort to speed up rule compaction, Dixon *et al.* developed CRA2 which focused both on speed and minimal rule population size (Dixon et al., 2003). CRA2 yielded similar performance to CRA but ran much faster when tested on the same dataset. Other similarly themed LCS rule compaction strategies include an approach for continuous-valued problem spaces (Wyatt et al., 2004), an approach for online rule compaction (Gao et al., 2006), an approach for clustering in XCS (Tamee et al., 2007), an approach which adds entropy calculation (Kharbat et al., 2008), and an approach designed for fuzzy rule representations (Shoeleh et al., 2011).

Recent efforts to move away from manual rule inspection and instead adopt a global pattern approach to knowledge discovery in LCSs have also shifted the priorities of rule compaction. In the context of complex noisy problem domains, it becomes impractical to expect rules to achieve a balance of maximum accuracy and generality. Therefore, placing too much emphasis on reducing the size of the rule population can lead to a loss of rule diversity and result in a reduction in overall performance. In (Urbanowicz et al., 2012a) a global approach to knowledge discovery is considered which introduced global evaluation statistics, and visualization strategies to achieve knowledge discovery in complex problem domains without manual rule inspection. This work focused on modeling noisy, complex patterns in supervised learning problems. In the present study we explore rule compaction somewhat differently than in previous efforts. First, we focus on LCS algorithms designed to address supervised learning domains. Specifically, while previous compaction strategies were designed to function within the context of XCS (Wilson, 1995), a reinforcement learning based LCS, we examine how rule compaction functions in supervised learning based LCSs such as UCS (Bernadó-Mansilla and Garrell-Guiu, 2003). Second, we approach rule compaction assuming that knowledge discovery will be achieved using a global approach as opposed to manual rule inspection. Lastly, we expand our evaluation of LCS performance beyond run time, training, and testing accuracy to consider the impact of rule compaction on an LCS's power to correctly prioritize predictive attributes and discover complex patterns of association as described in (Urbanowicz et al., 2012a). This expansion evaluates the compacted population's ability to yield successful knowledge discovery, as opposed to just successful classification.

The three rule compaction strategies introduced in this work focus on preserving or increasing the quality of the the final rule population rather than emphasizing a human readable population size. The first strategy, Quick Rule Compaction (QRC), is inspired by the match-covering mechanism in the third stage of CRA (Wilson, 2002). The sec-

ond strategy, Parameter Driven Rule Compaction (PDRC) is largely based on Dixon's approach. Specifically, for each training instance, PCRC finds the classifier in the correct set with the largest product of accuracy, numerosity, and generality and preserves that rule in the final population. The third strategy, Quick Rule Filter (QRF) is more of a rule filter than a compaction algorithm. It simply removes any rule in the population that does not have an accuracy above 0.5. The accuracy of a rule is the frequency of correct prediction for the subset of data instances the rule matches. We compare the resulting rule population performance following that application of our new approaches to three existing approaches (Fu's two approaches and CRA2). Changes in performance statistics are evaluated relative to rule populations without any rule compaction. We consider the advantages and disadvantages of each.

## Methods

In this section we describe (1) the LCS algorithm and run parameters used in this investigation, (2) the six rule compaction strategies considered in this study and (3) the experimental evaluation including data simulation, statistical analysis, and visualization.

### LCS Algorithm

We begin with a brief review of LCS algorithm concepts critical to understanding rule compaction. LCSs make class predictions based on "votes" made by rules which are relevant to a given instance from the dataset. Each rule possesses a condition and a classification. For example consider a hypothetical rule (0#1## - 1). The '#' serves as a wild card. This rule would match an instance from the data that looks like (02100 - 0), but not one that looks like (12100 - 0). Notice that this first instance example matches the rule, but the rule has incorrectly predicted the class to be '1', when in fact the class of this instance is '0'. These matching rules form what is known as a match set (i.e. the subset of rules in the population which match the attribute states of the dataset instance.) All rules that both match the instance as well as make the correct prediction form a correct set. During supervised LCS learning, when a rule is included in both a match and correct set, it's accuracy and fitness will increase, while if it is only involved in a match set (i.e. it matches but makes an incorrect classification) it's accuracy and fitness will decrease. For an in-depth review of LCS algorithms and how they function we refer readers to (Urbanowicz and Moore, 2009).

In order to evaluate each rule compaction strategy within a complex, noisy bioinformatics problem domain, we used an expanded Python encoding of AF-UCS (Urbanowicz et al., 2012b). AF-UCS (attribute feedback UCS), is an expanded and modified implementation of UCS (Bernadó-Mansilla and Garrell-Guiu, 2003) which incorporates a form of memory which feeds back into the genetic algorithm during

learning. UCS, or the sUpervised Classifier System, is an M-LCS based largely on the popular XCS algorithm (Wilson, 1995) but replaced reinforcement learning with supervised learning. UCS was designed specifically to address single-step problems such as classification and data mining, where delayed reward is irrelevant, and showed particular promise when being applied to epistasis and heterogeneity in (Urbanowicz and Moore, 2010). This expanded version of AF-UCS incorporates expert knowledge covering as described in (Urbanowicz et al., 2012c) to speed up learning.

The selection of run parameters for this evaluation was arbitrary. We adopted mostly default M-LCS run parameters. Parameters unique to this study include: 200,000 learning iterations, a rule population size of 2000, a rule generality of 0.75 in covering, tournament selection, uniform crossover, and subsumption on. The implementation described above is available on request ([ryanurbanowicz@gmail.com](mailto:ryanurbanowicz@gmail.com)) and will be posted on the LCS and GBML Central webpage.

### Rule Compaction Algorithms

The first LCS rule compaction strategy was CRA, a 3-stage algorithm aimed at dramatically reducing the number of rules and producing a human-readable ruleset (Wilson, 2002). Separate work modified the strategy for rule sorting in the first two stages of CRA such that less well trained classifiers seeking to model noisy problems could be taken into account (Fu and Davis, 2002). Two related strategies from this work are referred to throughout this paper as Fu1 and Fu2. Due to the inherent noise and complexity of our target problem domain, we have chosen to evaluate the two most successful approaches proposed by Fu and Davis as well as Dixon's CRA2, which yields similar performance but runs much faster (Dixon et al., 2003). These implementations are used as a baseline of comparison for our own proposed rule compaction strategies. In the following subsections we describe each of the six rule compaction algorithms considered, three existing strategies (Fu1, Fu2, and CRA2) as well as our three proposed strategies (QRC, PDRC, and QRF). Each have been implemented within the LCS algorithm described in the previous section, coded in Python.

**Strategy 1: Fu1** Fu's first approach is summarized as:

**Stage1:** Sort the rules by ascending numerosity (i.e. the number of copies of an identical rule). Begin with the first rule in the list, eliminate it and test the performance of the rest rules. If the performance becomes better or unchanged, continue to remove the next rule. If the performance is worse, reinsert that classifier to the ruleset and proceed to stage 2.

**Stage2:** Continue deleting each rule orderly from the ruleset, evaluating the performance of the remaining rules. If the performance is reduced after deletion, move that rule to the new ruleset. The deleted rule is

not considered in the next round of evaluation. After all rules being tested in such way, pass the new ruleset with all the rules causing performance reduction to the next stage.

**Stage3:** Calculate the number of instances in the training set a rule matches, move the rule matching most instances to the final ruleset, and delete instances matched to that rule from training set. Repeat the above three steps until the training dataset is empty or no rules match the remaining instances.

**Strategy 2: Fu2** Fu's second approach preserves the first two stages, while modifying stage 3 to take performance into consideration. The third stage can be described as follows:

**Stage3:** First sort the list of rules obtained from stage 2 by numerosity in increasing order, remove the last rule and evaluate the performance of the remaining classifiers. If the performance drops, reinsert that rule to the top of the list, thus it is involved in the following evaluation. Repeat such test on all classifiers and the final ruleset is composed of classifiers left in the list.

**Strategy 3: CRA2** Dixon's CRA2 approach avoids the need for step-wise performance evaluations. The basic idea of CRA2 is to identify the most useful rule for each instance in the training data. CRA2 examines each instance in the training data and builds a correct set from the rule population. For each instance, the most 'useful' classifier is marked for preservation in the final rule population. Dixon's approach determines the most useful rule to be the one in the correct set with the highest mathematical product of accuracy and numerosity. The original CRA2 was implemented in XCS, a reinforcement learning LCS, in which the correct set is called an action set. In the context of our implementation and evaluations, this is only a semantic difference.

**Strategy 4: Quick Rule Compaction (QRC)** Preliminary observations indicated that the third stage of Fu's first approach was the main cause of performance drop. QRC modifies this stage by using fitness instead of the number of instances a rule matches to retain useful rules. Additionally, QRC completely removes the first two stages utilized in both Fu1 and Fu2, eliminating the need for incremental performance evaluations of the whole rule population. Note that QRC ranks rules by fitness only once at the beginning of rule compaction. This ranking is not updated following the subsequent removal of instances. This differs from the original match-covering mechanism with the intention focusing on globally high fitness and the reduction of run time. Also, it is worth pointing out that in the LCS algorithm used, rule fitness is equal to rule accuracy. Pseudo-code for QRC is given in Algorithm 1.

```

Sort the rules decreasingly by fitness (or accuracy);
while Training dataset is not empty do
    MatchCount = 0;
    for Each instance in the training dataset do
        Determine whether it matches the first rule;
        if It matches then
            MatchCount ++;
        else
            Move the instance to new training set;
        end
    end
    if MatchCount > 0 then
        Copy the first rule to the final set;
    end
    Update training set to new training set;
    Delete the first rule from the sorted ruleset;
end

```

**Algorithm 1:** Quick Rule Compaction

#### Strategy 5: Parameter Driven Rule Compaction (PDRC)

Our PDRC approach is quite similar to Dixon’s speedy CRA2 implementation. In preliminary work we explored three different rule parameters (accuracy, numerosity, and generality) in an attempt to find the best numerical way to capture the utility of a rule. We considered them separately, the product of pairs, and the product of all three together. Keeping in mind that CRA2 utilized the product of accuracy and numerosity, we found that the product of accuracy, numerosity and generality performed best. Pseudo-code for PDRC is given in Algorithm 2.

```

for Each instance in the training dataset do
    Create MatchSet;
    Create CorrectSet;
    Find the classifier with highest product of accuracy,
    numerosity and generality in the CorrectSet;
    if The classifier is in final set then
        Pass;
    else
        Add the classifier to the final set;
    end
end

```

**Algorithm 2:** Parameter Driven Rule Compaction

**Strategy 6: Quick Rule Filter (QRF)** Our last proposed approach (QRF) is simply a filter which scans the rule population and deletes any rule with an accuracy  $\leq 0.5$ . This is intended to remove rules that predict class no better than by random chance at the time when learning was halted. Additionally, a rule is also deleted if it covers (i.e. matches) less than two instances in the dataset. This removes rules that are likely to be blatantly overfitting training instances. However,

such a deletion is prevented if the rule in question specifies only a single attribute. This accounts for the possibility of a rare variant, (i.e. a rare attribute state), which may be useful to preserve when seeking to interpret the rule population.

## Experimental Evaluation

**Data Simulation** Consistent with the nature of our target noisy bioinformatics problem of interest, we have applied our LCS algorithm in conjunction with all rule compaction strategies to a large set of simulated datasets which concurrently model heterogeneity and epistasis as they might appear in a SNP gene association study of common complex disease Urbanowicz and Moore (2010); Urbanowicz et al. (2012b,c). All data sets were generated using a pair of distinct, two-locus epistatic interaction models, both utilized to generate instances (i.e. case and control individuals) within a respective subset of each final data set. Each two-locus epistatic model was simulated without Mendelian/main effects, as a penetrance table as in Urbanowicz and Moore (2010). Due to the computational demands of LCSs, this study limited its evaluation to 3 heterogeneity/epistasis model combinations. For simplicity the minor allele frequency of each predictive attribute was set to 0.2, a reasonable assumption for a common complex disease SNP. The three model combinations included a pair of models with a heritability of either (0.1, 0.2, or 0.4). We considered model architectural “difficulties” of both “easy” and “hard” Urbanowicz et al. (2012d). Balanced datasets simulated from these models were generated as having four different sample sizes (200, 400, 800, or 1600) and a heterogeneous mix ratio of either (50:50 or 75:25) (e.g. 75% of instances were generated from one epistatic model, and 25% were generated from a different one). Twenty replicates of each dataset were analyzed and 10-fold cross validation (CV) was employed to measure average testing accuracy and account for over-fitting. Together, a total of 48 data set configurations (3 Model Combos x 4 Sample Sizes x 2 Ratios x 2 Difficulties), and a total of 960 data sets (20 random seeds each) were simulated. With 10-fold CV, 9600 runs the AF-UCS-based algorithm were completed followed by the same number of runs for each of the six compaction strategies.

**Statistical Analysis** For each run we track training accuracy, test accuracy, rule generality, macro population size, micro population size, and the run time required for rule compaction. Unlike previous investigations of rule compaction, we also consider three power estimates: (1) the power to find both heterogeneous underlying models, (2) the power to find at least one underlying model, (3) and the power to correctly rank attribute co-occurrence (Urbanowicz et al., 2012a). Power indicates the user’s ability to reliably mine knowledge from the evolved rule population. Co-occurrence power is a reflection of our ability to distinguish heterogeneous models from epistatic interactions. Results



over 10-fold CV are averaged. Statistical comparisons were made using the Wilcoxon signed-rank tests due to a lack of normality in the value distributions. All statistical evaluations were completed using R. Comparisons were considered to be significant at  $p$ -value  $\leq 0.05$ .

In order to further characterize differences between rule compaction approaches, we examine which specific rules remain following compaction. This similarity score is calculated as the ratio of rules preserved in two compacted rule sets to the size of the smaller rule set. Similarity scores were averaged over 10 CV runs.

**Visualization** Part of the global knowledge discovery process includes the application of intuitive visualizations to identify patterns of association within the rule population. In the present study we generate heat-maps to visualize compacted rule populations as described in (Urbanowicz et al., 2012a). Rules are encoded such that any specified attribute is coded as a 1 and any generalized attribute (#) is coded as a 0. Rule populations are visualized as a micro population, which means there are  $N$  copies of each rule reflecting respective numerosity ( $N$ ). The last processing step before visualization involves the application of a clustering algorithm to the encoded and numerosity expanded population of rules. In this study we employed agglomerative hierarchical clustering based on pearson correlation. For rules or attributes having undefined pearson correlation due to uniform values, 0 is assigned for the purpose of visualization. Both clustering and 2D heat-map visualization are implemented in R using *hclut* and *gplots* packages respectively. In this paper, we generate example visualizations using arbitrarily chosen datasets with a sample size of 1600, minor allele frequencies 0.2, model heritabilities of 0.2, a “Hard” model difficulty, and a heterogeneity ratio of 50:50.

## Results and Discussions

Table 1 gives a summary of the metrics evaluating rule populations following compaction. These are compared to the original rule populations prior to any form of compaction (NONE). Three existing approaches were considered (i.e. Fu1, Fu2, and CRA2) as previously described, as well as three approaches that have been proposed here (i.e. QRC, PDRC, and QRF). The color coding within table 1 makes it simple to quickly identify strategies which suffered significant losses, earned significant gains, or maintained performance relative to the original rule population. In reviewing the results, keep in mind that our proposed QRC strategy was most closely related to Fu’s approaches (Fu and Davis, 2002), and our proposed PDRC strategy was most closely related to CRA2. In this work we are most concerned with preserving or, if possible, improving the performance of a rule population while simultaneously seeking to reduce the overall rule population size.

We begin by discussing some of the more obvious trends. All approaches were successful at significantly reducing both the macro population size (i.e. the number of unique rules in the rule population) and the less interesting micro population size (i.e. the number of rules in the population, taking rule numerosity into account). The two Fu approaches yielded the most dramatic decrease in macro population size, followed by our own PDRC and CRA2. As might be expected, our QRF approach resulted in the deletion of only a handful of rules. Next, with the exception of QRC, all strategies significantly increased average rule population generality (i.e. portion of attributes that were generalized using a wild card within a given rule). This metric alone does not tell us much about performance, but all other metrics being equal, it is desirable to have a rule population that is maximally generalized. Given that within our simulated datasets, 20% of the attributes were predictive, wherein only 10% of attributes are predictive for a given heterogeneous dataset instance, we expect that the ideal rule generality would fall between 0.8 and 0.9. Turning our attention to rule compaction time, both Fu strategies took by far the most time to complete, owing to repeated accuracy evaluations of the rule population as a whole. Both CRA2 and our own PDRC strategy yielded similar, dramatically shorter run times. Our QRC approach further reduced run time by an additional order of magnitude, and lastly our QRF approach reduced run time even further by two additional orders of magnitude. Further statistical comparisons indicated that all differences in population size, generality and run time between all strategy pairs were also significant ( $p$ -value  $< 0.001$ ).

Next we examine the impact of each strategy on rule population accuracy and power. Keep in mind that power is a reflection of the algorithm’s success at prioritizing predictive attributes and patterns for successful knowledge discovery. Fu’s first strategy (Fu1) yielded the most significant dramatic loss in both accuracy and power (an apparent trade off with also generating the smallest rule population by far). Differently, Fu’s second strategy (Fu2) instead yielded significant increases in accuracy, but still resulted in modest, significant performance loss within all three power metrics. Comparing this to our proposed QRC strategy (most closely related to Fu1 and Fu2), we similarly observe a significant increase in accuracy (with a particularly notable increase in testing accuracy). Additionally, QRC results in a less significant and less dramatic loss in “Both-Power” and “Co-occurrence Power” when compared to the Fu’s approaches, as well as a small non-significant increase in “Single Power”. Considering our multiple evaluation objectives, our QRC approach appears to outperform both of Fu’s approaches. Interestingly when examining rule population similarity, only 45.51% of Fu1 and 56.55% of Fu2 unique rules were also found in the QRC rule population. Originally, we expected that the few rules left after running



Statistics (Averaged)	Existing Strategies				Proposed Strategies		
	NONE	Fu1	Fu2	CRA2	QRC	PDRC	QRF
Train Accuracy	0.7865	0.7085**	<b>0.8056**</b>	0.7938**	0.8027**	0.7966**	0.7873**
Test Accuracy	0.6189	0.5901**	0.6193**	0.6141**	<b>0.6293**</b>	<b>0.6160**</b>	0.6199**
Both Power	0.2729	0.0854**	0.2167**	<b>0.3052**</b>	0.2604*	0.2771**	0.2729
Single Power	0.7896	0.6969**	0.7552**	0.7781*	<b>0.7927</b>	0.7802	0.7906
Co-Occurrence Power	0.2802	0.0875**	0.2208**	<b>0.2896</b>	0.2667*	0.2854	0.2813
Generality	0.7831	<b>0.8392**</b>	0.7935**	0.7977**	0.7714**	0.7928**	0.7836**
Macro Population	276.46	<b>13.17**</b>	27.22**	78.54**	103.44**	60.30**	252.10**
Micro Population	2000	<b>208.52**</b>	414.91**	820.38**	1010.85**	972.33**	1971.53**
Rule Compaction Time (min)	0	4.9625	5.2235	0.0243	0.0064	0.0239	<b>1.60</b> $\times 10^{-5}$

Table 1: A comparison of population characteristics following rule compaction. Each statistic is averaged over all 960 simulated datasets and compared with populations prior to rule compaction (i.e. NONE) to determine statistical significance using the Wilcoxon signed rank sum test. (\*  $p$ -value  $< 0.05$ , \*\*  $p$ -value  $< 0.001$ ). Statistics in yellow have been significantly improved from NONE, while those in red have been significantly impaired. Time values in yellow are relatively fast, while those in red are comparatively slow. For each statistic, the the top finding across all strategies is given in bold.

Fu’s strategies would mostly be included in the larger QRC rule population. It turned out that half of the rules in Fu’s compacted rulesets did not possess a particularly high fitness relative to rules in the original rule population.

Examining CRA2, we observe a very small but significant reduction in both testing accuracy, and “Single Power”. However, CRA2 notably yielded a relatively large and significant improvement in “BothPower”, while maintaining “Co-occurrence Power”. Overall, CRA2 performance was certainly the best of the three existing strategies examined. Comparing CRA2 to our proposed PDRC strategy (again most closely related), we observe that PDRC also yields a small significant loss in testing accuracy. However, average testing accuracy for PDRC is significantly higher than for CRA2 ( $p$ -value  $< 0.05$ ). PDRC also yields a significant increase in “BothPower”, however it is not as large an improvement as with CRA2. PDRC does not significantly impact either “Single Power” or “Co-occurrence Power”. Overall, PDRC better preserves accuracy while also yielding a smaller average macro rule population than CRA2. Note that which micro population is smaller for PDRC than in CRA2, the opposite is true for micro population size. This indicates that PDRC is preferentially selecting rules with a higher average numerosity than CRA2. When examining rule population similarity, 74.89% of the unique rules commonly exist in both CRA2 and PDRC rule populations. This demonstrates CRA2 and PDRC tend to more frequently preserve the same rules within the rule population. It is different from what we observed comparing Fu’s approaches to QRC.

Lastly we examine the performance of QRF. QRF is the fastest strategy, and the only one which both significantly improves accuracy (although not as much as QRC and PDRC) as well as maintaining “BothPower” and slightly improving “Single Power” and “Co-occurrence Power” (how-

ever this improvement is not significant). The most obvious drawback to QRF is that it has very little effect on population size. However in the context of global knowledge discovery strategies, this may be of little importance.

Clearly there are strengths and weaknesses to each of the proposed approaches. For those interested in manual rule inspection with emphasis on obtaining the smallest rule set possible, it would appear that Fu’s approaches are best. If speed and the predictive ability of the rule population are priorities, the results suggest that QRC achieves the largest improvement in testing accuracy and runs approximately 1000 times faster than Fu’s strategies or approximately 10 times faster than CRA2 or PDRC. If the reader is interested in the most well rounded approach, PDRC yields the smallest rule population next to the Fu’s approaches and preserves or even improves all the metrics except a minor reduction in testing accuracy. Finally, if the reader is interested in the fastest approach that has the added benefit of completely preserving accuracy and power, but also does the least to reduce the size of the rule population, we suggest QRF.

## Visualization Results

Figures 1-3 present heat-maps visualizing rule population prior to rule compaction, following Fu2 compaction, and following PDRC compaction, respectively. Each row represents a rule while each column represents an attribute (X0-X19). Yellow blocks indicate attributes which have been specified within a rule, while blue indicates generalization (i.e. don’t care). Within each illustration, only four attributes were modeled as predictive. One epistatic model includes attributes (X0 and X1) and the second independent model includes attributes (X2 and X3). An accurate representation of this underlying model would yield rules that concurrently specify either attribute pair. Notice in Figure 1 that while there is clearly some noise, two distinct yellow

bands, correctly corresponding to (X0,X1) and (X2,X3), are apparent. Additionally hierarchical clustering most strongly links these individual attribute pairs.

In contrast to this successful interpretation of the underlying simulated patterns, Figure 2 shows the visualization of the same rule population following Fu2 compaction. Note that there are far fewer rules in this population, thus the respective row heights are larger than in Figure 1 since fewer rules are forced into the same figure height. Also, notice that while attributes (X2,X3) still form an obvious yellow band, (X0,X1) are no longer as clustered together as strongly, and the overall correct pattern is lost. The correct interpretation of this visual pattern should correspond with power estimates given in Table 1. An example of an effective compaction attempt which maintains, or perhaps, improves the interpretability of the heatmap is given by Figure 3. This figure shows the visualization of the rule population following application of PDRC to the original rule population. Notice that strong clusters are maintained between (X0,X1) and (X2,X3) attribute pairs, and some of the noise is eliminated when compared to Figure 1. Additionally, keep in mind that there are significantly fewer rules in the population visualized in Figure 3 than in Figure 1. It is worth noting that while not shown here, a similar visualization of the rule population following CRA2, yields a similarly clear illustration of patterns modeled in the dataset as seen in Figure 3.

### Conclusions and Future Work

In this study, we evaluate rule compaction algorithms for learning classifier systems (LCSs) with the goal of preserving or improving performance while reducing the size of rule population and facilitating knowledge discovery. Specifically we are most interested in applying rule compaction to problems domains with complex and noisy patterns. In addition, based on new global strategies for pursuing knowledge discovery within an LCS rule population (Urbanowicz et al., 2012a), we evaluate rule compaction performance by prioritizing global interpretation rather than traditional manual rule inspection. We also seek to avoid the most obvious shortcoming of most compaction strategies, i.e. computational complexity. We have introduced two new rule compaction strategies (QRC and PDRC) along with a rule filtering strategy (QRF), and compared them to three existing rule compaction methodologies (Fu1, Fu2, and CRA2) using a broader set of performance criteria than previously considered in similar studies.

The results highlighted the strengths and weaknesses of each rule compaction strategy in the context of our complex, noisy problem domain. Fu1 and Fu2 took the longest to run, yielding the smallest rule populations at the expense of accuracy and power. These strategies may be best suited to easier problems, or in problems where manual rule inspection is preferred. Our QRC strategy ran about 1000 times faster, achieving the highest testing accuracy, but re-

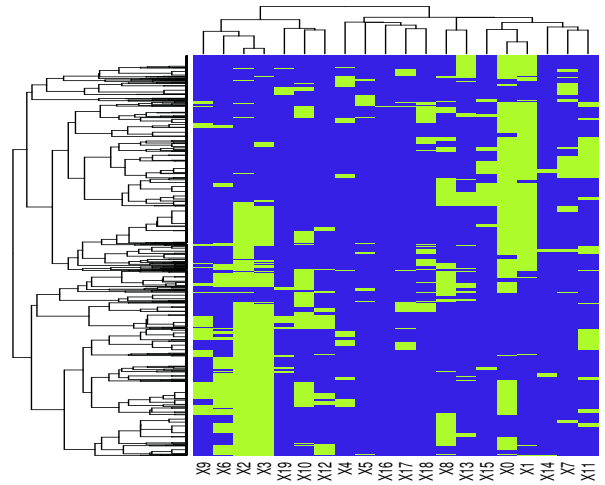


Figure 1: Rule population before compaction.

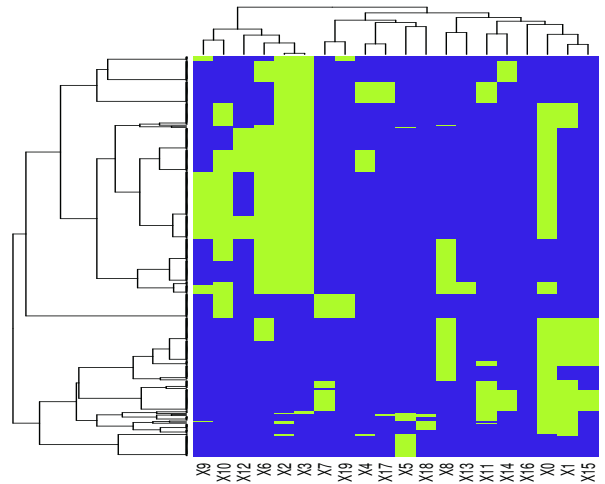


Figure 2: Rule population after Fu2 compaction.

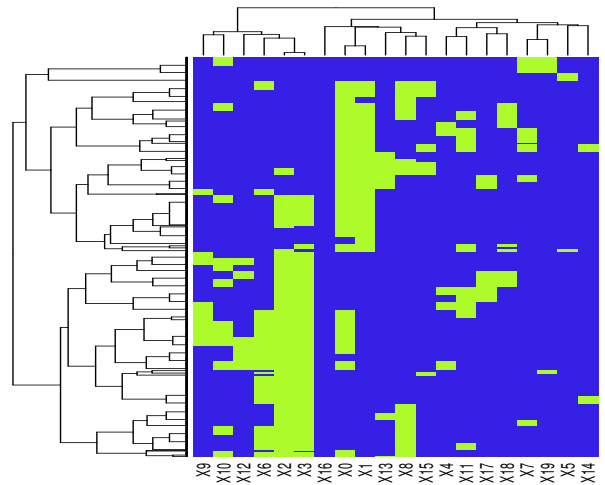


Figure 3: Rule population after PDRC compaction.

sulted in some minor losses in power. This strategy may be ideal when a very large dataset is being investigated or when the goal is classification rather than data mining. Overall our PDRC strategy yielded the most well rounded performance, with CRA2 close behind. Therefore, out of the six strategies examined, PDRC would be the best choice if a researcher wishes to effectively reduce the rule population size while preserving the overall performance of the rule population. Lastly QRF was the only approach that completely preserved or improved accuracy and power. This approach was also approximately 100000 times faster than Fu's approaches. However, this strategy did little to reduce the overall population size. This strategy would be best for situations in which rule population size is not a concern, but preservation of population performance is critical.

Future work will focus on exploring the constitution of rule sets after applying different rule compaction algorithms. Since there are distinct similarities between the compaction strategies, it would be interesting to further investigate the overlapping rules between populations formed by different strategies and better characterize the makings of an essential classifier. Also, while we have utilized a single specific supervised LCS, we would expect that other LCS implementations would similarly benefit from these proposed compaction strategies, as well as global strategies for knowledge discovery. We expect to utilize these compaction strategies in various real-world analyses and LCSs in future work.

### Acknowledgements

This work was supported by the William H. Neukom 1964 Institute for Computational Science and NIH grants LM011360, LM009012 and LM010098. RJU was supported by NIH grant R25 CA134286.

### References

- Bernadó-Mansilla, E. and Garrell-Guiu, J. (2003). Accuracy-based learning classifier systems: models, analysis and applications to classification tasks. *Evolutionary Computation*, 11(3):209–238.
- Dixon, P., Corne, D., and Oates, M. (2003). A ruleset reduction algorithm for the xcs learning classifier system. *Learning Classifier Systems*, pages 20–29.
- Fu, C. and Davis, L. (2002). A modified classifier system compaction algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 920–925. Morgan Kaufmann Publishers Inc.
- Gao, Y., Wu, L., and Huang, J. (2006). Ensemble learning classifier system and compact ruleset. *Simulated Evolution and Learning*, pages 42–49.
- Holland, J. H. (1986). A mathematical framework for studying learning in classifier systems. *Physica D*, 2(1-3):307–317.
- Kharbat, F., Odeh, M., and Bull, L. (2008). A new hybrid architecture for the discovery and compaction of knowledge from breast cancer datasets.
- Shoeleh, F., Hamzeh, A., and Hashemi, S. (2011). Towards final rule set reduction in xcs: a fuzzy representation approach. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 1211–1218. ACM.
- Tamee, K., Bull, L., and Pinngern, O. (2007). Towards clustering with xcs. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1854–1860. ACM.
- Urbanowicz, R., Granizo-Mackenzie, A., and Moore, J. (2012a). An analysis pipeline with statistical and visualization-guided knowledge discovery for michigan-style learning classifier systems. *Computational Intelligence Magazine, IEEE*, 7(4):35–45.
- Urbanowicz, R., Granizo-Mackenzie, A., and Moore, J. (2012b). Instance-linked attribute tracking and feedback for michigan-style supervised learning classifier systems. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, pages 927–934. ACM.
- Urbanowicz, R., Granizo-Mackenzie, D., and Moore, J. (2012c). Using expert knowledge to guide covering and mutation in a michigan style learning classifier system to detect epistasis and heterogeneity. *Parallel Problem Solving from Nature-PPSN XII*, pages 266–275.
- Urbanowicz, R. and Moore, J. (2009). Learning classifier systems: a complete introduction, review, and roadmap. *Journal of Artificial Evolution and Applications*, 2009:1.
- Urbanowicz, R. and Moore, J. (2010). The application of michigan-style learning classifiersystems to address genetic heterogeneity and epistasisin association studies. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 195–202. ACM.
- Urbanowicz, R. J., Andrew, A. S., Karagas, M. R., and Moore, J. H. (2013). Role of genetic heterogeneity and epistasis in bladder cancer susceptibility and outcome: a learning classifier system approach. *Journal of the American Medical Informatics Association*.
- Urbanowicz, R. J., Kiralis, J., Fisher, J. M., and Moore, J. H. (2012d). Predicting the difficulty of pure, strict, epistatic models: metrics for simulated model selection. *BioData mining*, 5(1):1–13.
- Wilson, S. (1995). Classifier fitness based on accuracy. *Evolutionary computation*, 3(2):149–175.
- Wilson, S. (2002). Compact rulesets from xcsl. *Advances in Learning Classifier Systems*, pages 65–92.
- Wyatt, D., Bull, L., and Parmee, I. (2004). Building compact rulesets for describing continuous-valued problem spaces using a learning classifier system. *Adaptive Computing in Design and Manufacture VI. Springer*, pages 235–248.

# Empowerment and State-dependent Noise - An Intrinsic Motivation for Avoiding Unpredictable Agents

Christoph Salge<sup>1</sup>, Cornelius Glackin<sup>1</sup> and Daniel Polani<sup>1</sup>

<sup>1</sup>University of Hertfordshire, College Lane, Hatfield AL10 9AB, UK  
c.salge@herts.ac.uk

## Abstract

*Empowerment* is a recently introduced intrinsic motivation algorithm based on the embodiment of an agent and the dynamics of the world the agent is situated in. Computed as the channel capacity from an agent's actuators to an agent's sensors, it offers a quantitative measure of how much an agent is in control of the world it can perceive. In this paper, we expand the approximation of empowerment as a Gaussian linear channel to compute empowerment based on the covariance matrix between actuators and sensors, incorporating state-dependent noise. This allows for the first time the study of continuous systems with several agents. We found that if the behaviour of another agent cannot be predicted accurately, then interacting with that agent will decrease the empowerment of the original agent. This leads to behaviour realizing collision avoidance with other agents, purely from maximising an agent's empowerment.

## Introduction

One important and unique aspect of living organisms is how they generate their behaviour. Sims (1994) demonstrated that simple motivations can be enough to generate complex behaviour that evokes a resemblance of life. Ultimately all organisms are subject to evolution and their behaviour is a product or by-product of a process directed by reproductive fitness and survival. However, from a cognitive perspective, it seems difficult for an agent to always relate behaviour back to survival. From an evolutionary perspective, it is also questionable how the sparse sampling of random behaviours could lead to good solutions. Nature solves this problem with the development of behavioural proxies or motivations (Scott-Phillips et al., 2011), such as the ability to perceive and avoid pain, which produces behaviour considered beneficial for survival. In artificial life the corresponding research aims to identify, quantify and replicate these motivations.

Significant research interest has been directed at methods known as "intrinsic motivations", methodologies to generate behaviours for agents without the requirement of an externally specified reward or utility structure; importantly, they emerge exclusively from the agent-environment dynamics. Here, instead of a specific goal, the generation of behaviour

depends on an internal motivation. Most of them focus on learning and exploration, and try to quantify an organism's urge to understand its environment (Schmidhuber, 1991; Der et al., 1999; Steels, 2004; Prokopenko et al., 2006; Ay et al., 2008).

In this paper we focus on one of these methods, which is based on *empowerment* (Klyubin et al., 2008). Empowerment provides a "universal utility", i.e. a utility landscape over the state space of an agent which is defined purely by the agent-world dynamics. In contrast to other methods it does not focus on learning or exploration, but identifies preferable states in a known local environment. Empowerment considers the probabilistic map from a sequence of the agent's actions to a world state resulting from these actions as a channel; empowerment is then formally defined as the Shannon (1948) channel capacity of this channel. Essentially, empowerment is an information-theoretic generalization of the control-theoretic concept of controllability (Touchette and Lloyd, 2000).

The basic motivation behind the empowerment concept is that it is preferable to be in a state where the agent's actions have the largest influence on the perceivable world, or *Umwelt* (von Uexküll, 1909), of the agent. From an empowerment perspective, the ideal state to be in is one that offers a high number of choices that all lead to different outcomes that can be causally (and predictably) distinguished (i.e. controlled) by the agent. States to avoid are those where noise interferes with the influence of agent actions on its resulting state (lack of controllability), and those where the agent can only reach a low number of possible resulting states through its actions (lack of reachability). If we are dealing with a deterministic system, where each action leads to one specific outcome, then the criterion reduces to pure reachability. An example for the latter special case is given by Klyubin et al. (2005a) where he demonstrates the relationship between average distance and empowerment in a grid-world maze.

In this paper we continue prior work (Salge et al., 2012) which provides a fast approximation of empowerment using a Gaussian linear channel. Here, we expand this method to



allow empowerment computation based on the covariance matrix between actuators and sensors, which, importantly, allows for the incorporating of state-dependent noise. This provides us with an appropriate and fast empowerment algorithm for the study of continuous systems with several agents, which was not possible before. We will outline the modification for the approximation method, and then use it to examine the empowerment in a simple, continuous multi-agent system.

## Related Work

While many forms of adaptation and learning require some external goal-orientated supervision, critique, or perspective, it is now well understood, that a focus on embodiment (Pfeifer et al., 2007) provides a vehicle for self-determination which does not necessitate such external goals. Based on this, recent efforts have been made to understand agent control and agent motivation in the framework of the perception-action loop (Lungarella et al., 2005; Bertschinger et al., 2008), see also Fig. 1.

For example, Homeokinesis (Der et al., 1999) is a predictive methodology which adapts its perception-action loop on the fly, and drives an embodied agent to exploit its own embodiment to generate movement. It is related to other intrinsic motivation methods, such as “artificial curiosity” (Schmidhuber, 1991), or the “autotelic principle” (Steels, 2004), where the agent becomes self motivated when the challenges it faces are adequate for its skill level, inspired by the concept of “flow” (Csikszentmihalyi, 2000) for humans.

More recent developments use the framework of predictive information to produce intrinsically motivated robot control, generating behaviour only by looking at how specific actuator inputs change the agents sensor state (Prokopenko et al., 2006; Bialek et al., 2001; Ay et al., 2008). This is similar to the idea of empowerment, which is also fully defined by the channel between an agent’s actuators and sensors. One key difference though, is that empowerment provides a utility landscape by assigning a value to each state of the environment, where previous approaches are focussed on producing specific actions. Of special interest to the topic of this paper is also the work by Capdepuy et al. (2012), where he studies how empowerment is limited when several agents are using a joint channel, and what restriction this applies to agent coordination.

## Formalism

Given a perception-action-loop, as seen in Fig. 1, Klyubin et al. (2005a) defined empowerment as the channel capacity (Shannon, 1948) between an agent’s actuators  $A$  and sensors  $S$ , with the world being in state  $r \in R$ . Each state  $r \in R$  has its own empowerment value, which only depends on the

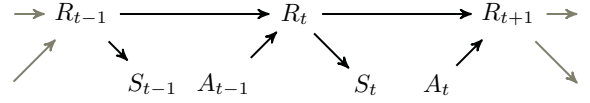


Figure 1: The perception-action-loop, unrolled in time  $t$ , visualised as a Bayesian network. The random variable  $S$  is the sensor of an agent;  $A$  is the actuator of an agent, and  $R$  represents the rest of the system.

channel between  $A$  and  $S$  in that state.

$$\mathfrak{E}(r_t) := C(p(s_{t+1}|a_t, r_t)) = \max_{p(a_t)} I(S_{t+1}; A_t | r_t). \quad (1)$$

Similarly, a sequence of  $n$  actions can be considered (called  $n$ -step empowerment), where the action-sequences are treated as a vector of random variables. The sensor state  $s \in S$  is then usually further in the future, and can also be a vector of random variables.

## Continuous Empowerment

Empowerment is defined for both discrete and continuous variables, but while it is possible to determine the channel capacity for the discrete case (for example by using the Blahut-Arimoto Algorithm (Blahut, 1972)), this is not generally possible for the continuous case. Jung et al. (2011) introduces a technique called Monte-Carlo Integration to approximate empowerment, but this method is very computationally expensive.

A faster method (described in detail by Salge et al. (2012)) to approximate the empowerment of a continuous channel is to treat it as a linear channel with added independent and identically distributed (i.i.d.) Gaussian noise.

$$S = TA + Z, \quad (2)$$

where  $S$  is an  $m$ -dimensional, continuous random variable,  $A$  is an  $n$ -dimensional, continuous random variable,  $T$  is a linear mapping, represented by a  $m \times n$  matrix, and  $Z_i \sim \mathcal{N}(0, N_i)$ , with  $i = 1, \dots, m$ , is another multi-dimensional, continuous, i.i.d. random variable, modelling isotropic noise in the sensor dimensions.

Assuming that there is a power constraint  $E(A^2) \leq P$  (without it the channel capacity would be arbitrarily large), this can be solved (Telatar, 1999) by applying a Singular Value Decomposition (SVD) to the transformation matrix  $T$ . The resulting singular values  $\sigma_i$  are then used to compute the channel capacity via the water-filling algorithm, as if this was a parallel Gaussian channel (Cover and Thomas, 1991). The channel capacity is then

$$C = \max_{P_i} \sum_i \frac{1}{2} \log(1 + \sigma_i^2 P_i), \quad (3)$$



where  $P_i$  is average power used in the  $i$ -th channel, following the constraint that  $\sum_i P_i \leq P$ . As the channel capacity achieving distribution is a Gaussian distribution, this means the optimal input distribution is a Gaussian with a variance of  $P_i$  for each channel.

### State-Dependent Noise

Salge et al. (2012) use simplifications that are only possible because the model's noise  $Z$  is assumed to be i.i.d distributed with a fixed variance. This forces the previous algorithm to assume the same level of noise for every state of the environment, and also makes it unable to model coloured, i.e. covariate, noise.

To address this problem we used the covariance matrix between actuators and sensors to capture the relationship between them, as well as the current noise level, and then reduced this problem to a parallel Gaussian Channel with i.i.d noise with the same capacity. First we chose  $n$  actuator variables  $A_1, \dots, A_n$ , and  $m$  sensor variables  $S_1, \dots, S_m$ . Now we determine the covariance matrix  $K$  between all these values. In our example, this is done by computing the pairwise covariance between sampled values for each of these variables. Alternatively, one could use the covariance function  $k(\cdot, \cdot)$  of a Gaussian Process that models the system to obtain the covariance matrix (not done in this experiment).

$$K = \begin{pmatrix} k(a_1, a_1) & \dots & k(a_1, a_n) & k(a_1, s_1) & \dots & k(a_1, s_m) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k(a_n, a_1) & \dots & k(a_n, a_n) & k(a_n, s_1) & \dots & k(a_n, s_m) \\ k(s_1, a_1) & \dots & k(s_1, a_n) & k(s_1, s_1) & \dots & k(s_1, s_m) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k(s_m, a_1) & \dots & k(s_m, a_n) & k(s_m, s_1) & \dots & k(s_m, s_m) \end{pmatrix} \quad (4)$$

$$= \begin{pmatrix} K_{a,a} & K_{a,s} \\ K_{s,a} & K_{s,s} \end{pmatrix}. \quad (5)$$

Now, if the variable  $A = A_1, \dots, A_n$  assumes a concrete value  $a = a_1, \dots, a_n$ , then this results in a specific, multi-variate Gaussian distribution for  $S = S_1, \dots, S_m$ , with

$$S = \mathcal{N}(\mu_s, K_s). \quad (6)$$

Note that  $K_s \neq K_{s,s}$ .  $K_s$  can be computed (Rasmussen and Williams, 2006) as:

$$K_s = K_{a,a} - (K_{s,a} K_{s,s}^{-1} K_{a,s}). \quad (7)$$

Assuming that the mean of actuator distributions of  $A$  is zero<sup>1</sup> we can also determine the mean for  $S$  given as a specific value of  $a$  as

$$\mu_s = K_{s,a} K_{s,s}^{-1} a. \quad (8)$$

We see from Equ. 7 that the covariance only depends on the original covariance matrix, and not on the actual value

<sup>1</sup> If the mean of a distribution is not zero it can be shifted without affecting the mutual information

of  $a$ . Also, from Equ. 8 we see that the new mean of the distributions is a linear transformation of  $a$ , with the matrix  $K_{s,a} K_{s,s}^{-1} = T'$ . So, a variation of  $a$  affects the mean of the resulting distribution of  $S$ , but not its covariance.

As a result, the relationship between  $S$  and  $A$ , as modelled by the covariance matrix, can be expressed as a linear, multiple input, multiple output channel with added coloured noise as

$$S = T' A + Z', \quad (9)$$

with  $Z' \sim \mathcal{N}(0, K_s)$ . Note, that there is no approximation in this step, the linear channel fully captures the dynamics of the system that are still present in the covariance matrix.

This can be further reduced to a channel with i.i.d. noise. For this, note that rotation, translation and scaling operators do not affect the mutual information  $I(S; A)$ . We start by expressing  $Z'$  as

$$Z' = U \sqrt{\Sigma} Z V^T, \quad (10)$$

where  $Z \sim \mathcal{N}(0, I)$  is isotropic noise with a variance of 1, and  $U \sqrt{\Sigma} V^T = K_s$  is the SVD of  $K_s$ .  $U$  and  $V^T$  are orthogonal matrices, and  $\Sigma$  contains the singular values. The square roots of the singular values scale the isotropic noise to the right variance; the noise is then rotated to resemble the original coloured noise. Note, that all singular values have to be strictly larger than zero, otherwise there would be a channel in the system without noise, which would lead to infinite channel capacity. Thus, we can consider  $\sqrt{\Sigma}^{-1}$ , a diagonal matrix with entries which are the inverse of the singular values in  $\sqrt{\Sigma}$ . This allows us to reformulate:

$$S = T' A + U \sqrt{\Sigma} Z V^T \quad (11)$$

$$U^T S V = U^T T' A V + \sqrt{\Sigma} Z \quad (12)$$

$$\sqrt{\Sigma}^{-1} U^T S V = \sqrt{\Sigma}^{-1} U^T T' A V + Z \quad (13)$$

$$\sqrt{\Sigma}^{-1} U^T S = \sqrt{\Sigma}^{-1} U^T T' A + Z V^T \quad (14)$$

$$\sqrt{\Sigma}^{-1} U^T S = \sqrt{\Sigma}^{-1} U^T T' A + Z \quad (15)$$

The last step follows from the fact that the rotation of isotropic Gaussian noise is isotropic Gaussian noise. This reduces the whole problem to a MIMO channel with isotropic noise with the same channel capacity. We simply define the transformation matrix  $T$  used in  $S = T A + Z$  as

$$T = \sqrt{\Sigma}^{-1} U^T T', \quad (16)$$

and apply the solution outlined for the simpler channel. This reduction allows the fast approximation of empowerment based on the covariance matrix between actuators and sensors, which can be either obtained via sampling of the environment, or by relying on a Gaussian Process Learner. This allows us to model the actual noise present in different states of the environment, which is then represented in the modified  $T$ .

## Model

We now apply this new method to a simple, continuous multi-agent system. The other agents are introduced to provide a changing level of noise in the environment. The covariance based empowerment approximation allows us to study how an agent would deal with different levels of noise.

### World Model

The model is a continuous, flat, two-dimensional world, populated by circular agents. Each agent has a radius  $l = 0.15$  meters<sup>2</sup>. Each agent is defined by:

- a position, stored in a vector  $q$ , which contains a real-valued  $x$  and  $y$  coordinate,
- a speed  $\dot{q}$ , expressing the change in  $x$  and  $y$  per second,
- a direction  $d$  the agent is currently facing, measured in radians, where 0 means the agent is heading north.

### Actuation

At the beginning of each time step, the agent has the choice to turn within its turning radius, which is 45 degrees, or one-eighth of a circle. The agent chooses a real value between -1.0 (turning 45 degrees counter-clockwise) and 1.0 (turning 45 degrees clockwise); 0.0 means the agent maintains its current heading. The agent turns instantaneously and then, for the duration of the time step, the agent continuously accelerates at  $0.03 \text{ m/s}^2$ . In our model the agent will accelerate at full power, the only choice is the direction of acceleration.

### Simulation

The only other acceleration force that applies to agents arises from collisions with other agents. Whenever the distance between two agents becomes less than the sum of their radii, a collision occurs. This is modelled as an elastic collision, so the agents can come closer than this distance, but will be subject to linearly increasing acceleration away from the center of mass of the other agent. The acceleration from the collision for the first agent can be computed as

$$\ddot{q}_c = \max(0, |l_i + l_j| - |q_1 - q_2|) \cdot (q_1 - q_2) \cdot c, \quad (17)$$

where  $c$  is a constant that determines how hard the elastic collision is. For lower  $c$ , colliding agents move further into each other before they bounce apart. Furthermore, to keep the velocity of the agents limited, there is a constant amount of friction applied to the agents. At each time step agents lose 5% of their velocity.

The progress of the model through time is simulated by breaking each time step into 20 pieces of equal length, and for each of those an appropriate fraction of the acceleration

<sup>2</sup>For ease of notation the unit length will be called meter, and the length of a time step will be one second

of the agent is added to its speed, and then the speed is added to the agent's current position. This is equivalent to explicit Euler integration.

Note that this model allows slip, i.e. an agent can head in one direction (where it is also accelerating to), while moving in a different direction. Turning does not change the current inertial movement.

## Experiments

### Hypothesis

Preliminary observations of the agent's behaviour indicated that an increase in the chance of a future collision with other agents is accompanied by a reduction in the current empowerment value. Therefore, our hypothesis for this simulation is that since the behaviour of other agents cannot be predicted, they will act as a source of noise in the environment, and colliding with them would be detrimental for an agent's empowerment.

### Different Scenarios

To test this hypothesis, and evaluate it systematically, we set up three different scenarios. In each scenario there are two agents. For the first agent we measure the empowerment and collision chance at different starting coordinates, located between -1 and 1, both for the  $x$  and  $y$  coordinate. The first agent is always heading north-east, and starts with a speed of  $0.03 \text{ m/s}$  in that direction. The second agent is always located in position (0.5,0.5) and is heading south-west with a speed of  $0.03 \text{ m/s}$ . The three scenarios differ in the behaviour of the second agent:

**Unpredictable Agent:** The second agent chooses actions uniformly random at the beginning of each time step, turning within its possible turn radius. The choice of the second agent cannot be predicted by the first.

**Predictable Agent:** The second agent always chooses to maintain its current direction, i.e. it moves in a straight line. This is known to the first agent, and incorporated into its model.

**Immovable Agent:** The second agent is anchored to its position, essentially constituting a fixed obstacle. It still reflects other agents colliding with it.

Note that the term agent here is used loosely as a "catch-all" term for other objects in the environment, which could be agents, movable objects or just fixed obstacles.

### Measurements

We computed the 4-step empowerment for Agent 1 for the three scenarios, for different starting positions. So, the actuation variables  $a_1, \dots, a_4$  denote what action Agent 1 chooses at the beginning of the first, second, third and fourth

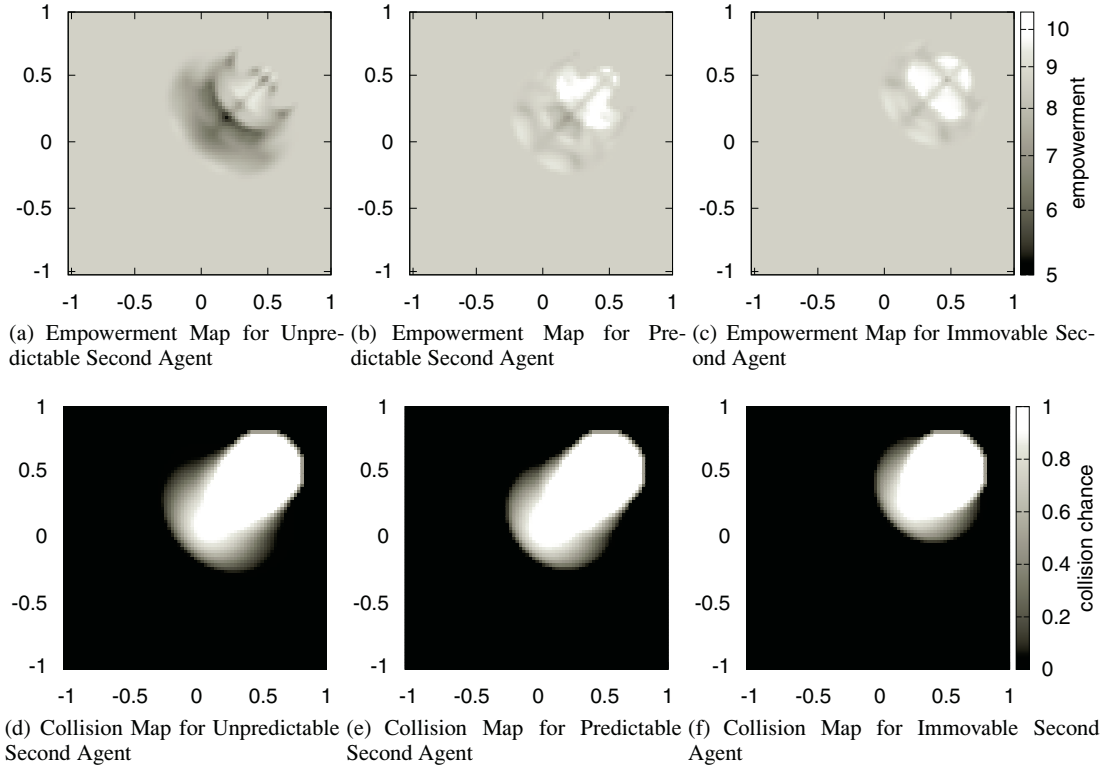


Figure 2: Plots of the empowerment and collision probability for different starting positions of the first agent. The Figs. 2(d)-2(f) show the fraction of action sequences that lead to a collision between the agents. Figs. 2(a)-2(c) show the empowerment of the first agent. The second agent is always located at position (0.5,0.5). Initial heading for the first agent is north-east, for the second agent it is south-west.

time step, respectively. The sensor input considered for empowerment were the values of  $x, y, \dot{x}, \dot{y}$  after the fourth time step, so the speed and position of the agent after the actuation sequence has been executed.

For each starting position for Agent 1, we used the actual simulation model to create an amount of samples, consisting of actuator variables and resulting sensor values. We used regular sampling, so that each time step the agent could only choose the 5 values of  $\{-1.0, -0.5, 0.0, 0.5, 1.0\}$ , leading to  $5^4 = 625$  possible action sequences in four steps. Each of these sequences was then simulated 10 times, leading to 6250 samples overall.

In the “unpredictable agent” scenario the action of the other agent was chosen uniformly random for those simulations, whereas in the predictable and unmovable agent scenario the simulation “knew” what the other agent would do, leading to a predictable outcome for each action sequence. So the 10 repeated samplings of the same action sequence only led to different results in the unpredictable agent scenario, since the pseudo-random generator would potentially chose different actions for the second agent. The resulting 8 times 6250 values were then used to pairwise compute a

covariance matrix between all 8 values (4 actuation values, 4 resulting sensor values), which was in turn used to compute the channel capacity from the actuation variables to the sensor values. This allowed us to compute empowerment of Agent 1 for different starting positions. We also recorded, for each starting position, what percentage of the sampled action sequences would lead to a collision with the second agent within the first four time steps.

## Results

The results demonstrate that colliding with the unpredictable agent leads to a substantial loss in empowerment, compared with the other scenarios. Consider first the collision maps depicted in Figs. 2(d)-2(f), which show what fraction of the action sequences in a given starting position leads to a collision between the agents. This segments the space of starting positions for agent one into three areas.

The area with zero collision probability are all locations where there is no chance for the agents to interact. These areas are thus of little interest for our central hypothesis. The empowerment landscape in these areas is constant, as expected for an unstructured environment. We will consider

this constant value to be the baseline value of empowerment for comparison.

The second area are those starting locations where the agents always collide. This is mainly a circle of diameter 0.3 *m* around (0.5,0.5), where the agents already start in collision, and a connected area where the agents start separated but are moving towards each other. Towards the center of the circle the agents overlap the most, and there are several areas of higher empowerment. This results from the specific collision mechanics in our simulation. As we model near-physical elastic collision, agents who overlap can be considered as “storing potential energy”, to apply a physical analogy. This high potential energy allows faster acceleration, which allows the agent to reach a greater variety of locations. The analogy here would be riding a bike, both on a flat surface, and starting on top of a hill. The extra speed gained riding downhill allows the rider to reach a greater variety of locations. Similarly, Agent 1 can control, in part, where this extra acceleration moves it, resulting in greater empowerment for Agent 1. We can also see that this effect is greater in Fig. 2(b) and Fig. 2(c), since the agent here can fully predict where this extra acceleration will lead it. In Fig. 2(a) this effect is less powerful, as the unpredictable movement of the second agent makes it harder to predict where the first agent will end up, thereby lowering the empowerment of Agent 1.

The most interesting area for our hypothesis are those locations where it is uncertain whether the agents collide or not. These areas are located further to the south-west of the second agent. So a first agent starting here is moving towards the second agent, but is far enough away that some actions might lead to an avoidance of collision. Now, the outcome of different action sequences not only depends on the first agent’s actions, but also on what the second agent does. If the second agent is unpredictable, i.e. moves at random, then the simulation done by the first agent will result in different outcomes for the same actions, introducing noise to the channel between actuators and sensors. This causes a measurable increase in uncertainty. The effect of this can be seen in Fig. 2(d). If we now compare this with the predictable second agent scenario, we see much less empowerment reduction with a predictable agent. While the second agent might still block access to some locations, it does not introduce noise into the outcome. Thereby, a collision with the second agent does not reduce the empowerment as much, as seen in Fig. 2(b). Similarly for the immovable agent, the empowerment here is only slightly reduced when an agent is on a collision course. So, the main cause for the drop of empowerment in our model is not the *collision* with a second agent, but the collision with an *unpredictable* agent.

### Empowerment Control

The difference between a predictable and unpredictable agent becomes even clearer if we look at the resulting agent

control. We implemented a greedy empowerment maximisation control. From five candidate actions (-1,-0.5,0,0.5,1) it picks the action that *leads to* the state with the largest empowerment value. For this, 4-step empowerment is calculated for all five states resulting from the candidate actions.

Fig. 3 shows the resulting trajectories of the first agent for different starting positions. Once a collision with the second agent occurs, the line becomes dashed. In both figures the first agent selects actions that maximise its empowerment for the next step; the only difference is the behaviour of the second agent. Both second agents start heading towards the first agent, but the second agent in Fig. 3(a) just moves straight, while the second agent in Fig. 3(b) moves at random. So in Fig. 3(b) the simulations of the first agent to determine the empowerment of possible future states cannot accurately predict the second agent. This means that the possibility to interact with the second agent becomes a source of noise, and empowerment maximisation avoids actions that lead to trajectories where the possibility of interaction with the second agent might arise. As a result, only three of the resulting trajectories collide with the second unpredictable agent. In the other case, shown in Fig. 3(a), empowerment sees no problem with colliding with the second agent, as it does not introduce noise into its action-perception channel and therefore it permits a lot of trajectories to end up in collisions.

### Discussion

The specific model we are considering here results in a collision avoidance behaviour regarding the second agent, if that agent is unpredictable. While it could be argued that other agents tend to be hard to predict, and therefore the assumption that other agents introduce noise is reasonable, we emphasize that the aim here was *not* to specifically produce obstacle avoidance. Also, the specific behaviour of empowerment depends on how the environment is modelled. If collisions result in loss of velocity, or even loss of actuation possibilities (like broken motors), then empowerment would avoid any collision. Even if this was not the case, if obstacle avoidance was desired, one could ‘spike’ the system dynamics as to make the agent believe that it would break down when it collides, which would induce strict obstacle avoidance. This would correspond to ‘programming’ or, rather, ‘nudging’ the empowerment-based behaviour engine towards desirable behaviours. Importantly, if there were an explicit goal of obstacle avoidance we would not contest that explicit obstacle avoidance algorithms would be superior. The advantage of an empowerment controlled agent lies in its universality. The same algorithm that avoids collision also balances an inverted pendulum (Jung et al., 2011), finds central positions in a maze (Klyubin et al., 2005b), can be used to adapt sensors (Klyubin et al., 2005a), and leads to the formation of patterns in multi-agent simulations (Capdepuuy et al., 2007).

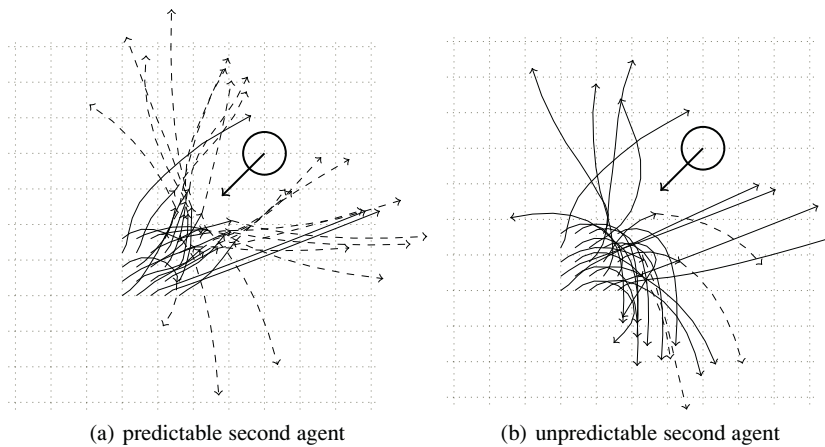


Figure 3: A comparison plot for the behaviour of an empowerment maximising agent. The arrows trace the trajectories of the empowerment controlled agent; each arrow is a different simulation, starting from a slightly different point. The initial heading of the empowerment agent is north-east. The circle indicates the starting position of the other agent, its initial heading is south-west. The lines become dashed if a collision between the agents occurs. Fig. 3(a) has a predictable second agent, which just moves straight ahead. Therefore, empowerment sees no need to avoid, and most trajectories lead to collisions. Fig. 3(b) has a second agent that chooses random actions. It is therefore a source of noise and the empowerment driven agent avoids colliding with it in most cases.

The point of using the intrinsic, empowerment-based behaviour is that it is more generic and grounded in the agent-system dynamics, and incorporates implicit difficulties that the agent may encounter. Thus, instead of imposing explicit conditions on when to activate a sub-behaviour, such as obstacle avoidance, one could incorporate desired hard behaviours, where required, into “surrogate” modifications of the physics of the system and let the empowerment-based behaviour engine generate the behaviours based on these modifications, whilst leaving the natural dynamics of the system unchanged for all other situations. Such an approach may be able to provide the agent with a more flexible repertoire of options, whilst respecting required hard constraints. It would also take a step towards “implicit programming”.

In terms of applications, it would also be interesting to see how an empowerment-based system would deal with a navigation task in a crowded environment, such as walking down Oxford Street at prime shopping time. In general, one should avoid colliding with people, but one could speculate that understanding how another person is going to move would allow an agent to operate closer to that person, with less chance of collision, and therefore less loss of empowerment. This would require the introduction of separate models for different agents, which would then allow an agent to model how predictable another agent is, and consequently adjust its behaviour towards different agents. Note that in this hypothetical example empowerment is clearly computed based on the agent model of the world. It does not matter how predetermined another agent’s behaviours are, but how well this can be predicted by the internal model.

More generally, we see that empowerment depends on the agent’s internal model of the world. Reducing the uncertainty in one’s internal model increases empowerment, which then raises the question, how suited is empowerment for exploration? We speculate that this depends on the horizon of the empowerment optimization. In the short term, interacting with another unpredictable agent will be detrimental, and avoiding it will preserve an agent’s empowerment. However, in the long term interacting with another agent might increase the predictability of said agent (by virtue of learning a better model of the agent), and this will improve the empowerment of the first agent on subsequent later interaction. This also indicates a more general distinction between different sources of noise in the environment; those that are unpredictable at first, but can be learnt, and those that are actually random.

## Future Work

In terms of robot control the development of both continuous empowerment (Jung et al., 2011) and fast continuous empowerment (Salge et al., 2012) was crucial to applying empowerment to real life systems in real time. The addition of state dependent noise now brings back the aspect of controllability to empowerment, and opens the possibilities for robotic control applications.

Imagine a robot that follows a human around in order to assist it. An empowerment map of the environment could provide the robot with an additional, supporting fitness function. Primarily, the robot would be interested in keeping its distance from the human. The reachability aspect of empow-



erment would keep it from getting stuck (actions would all lead to the same outcome), or ending up in a dead end. Controllability would keep the robot from getting too close to any human agent, considering they would be hard to predict. This could offer some additional incentives to the robot, once the primary objective is reached. So, instead of waiting, the robot could manoeuvre into a better position, where it could quickly get to a lot of other places, or where it would be less in danger of crossing paths with an unpredictable human agent.

## Conclusion

We demonstrated how state dependent, coloured noise can be integrated into the fast quasi-linear Gaussian approximation of empowerment. This allows this faster empowerment approximation to regain the state-dependent noise-sensitivity of the original formalism. The extension allows us to demonstrate with our examples how empowerment is negatively influenced by interacting with a local source of noise. We see that a greedy empowerment-maximising agent tries to avoid such interaction; in our specific case the collision with another agent. The same principle would apply to other forms of noise, i.e. other aspects of the environment that either cannot be, or have not been properly modelled by the agent.

## Acknowledgements

This research was supported by the European Commission as part of the CORBYS (Cognitive Control Framework for Robotic Systems) project under contract FP7 ICT-270219. The views expressed in this paper are those of the authors, and not necessarily those of the consortium.

## References

- Ay, N., Bertschinger, N., Der, R., Güttler, F., and Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *The European Physical Journal B-Condensed Matter and Complex Systems*, 63(3):329–339.
- Bertschinger, N., Olbrich, E., Ay, N., and Jost, J. (2008). Autonomy: An information theoretic perspective. *Biosystems*, 91(2):331–345.
- Bialek, W., Nemenman, I., and Tishby, N. (2001). Predictability, complexity, and learning. *Neural Computation*, 13(11):2409–2463.
- Blahut, R. (1972). Computation of channel capacity and rate-distortion functions. *Information Theory, IEEE Transactions on*, 18(4):460–473.
- Capdepuy, P., Polani, D., and Nehaniv, C. (2007). Maximization of potential information flow as a universal utility for collective behaviour. In *Artificial Life, 2007. ALIFE'07. IEEE Symposium on*, pages 207–213. IEEE.
- Capdepuy, P., Polani, D., and Nehaniv, C. (2012). Perception-action loops of multiple agents: informational aspects and the impact of coordination. *Theory in Biosciences*, pages 1–11.
- Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley-Interscience, 99th edition.
- Csikszentmihalyi, M. (2000). *Beyond boredom and anxiety*. Jossey-Bass.
- Der, R., Steinmetz, U., and Pasemann, F. (1999). *Homeokinesis: A new principle to back up evolution with learning*. Max-Planck-Inst. für Mathematik in den Naturwiss.
- Jung, T., Polani, D., and Stone, P. (2011). Empowerment for continuous agent environment systems. *Adaptive Behavior*, 19(1):16.
- Klyubin, A., Polani, D., and Nehaniv, C. (2005a). All else being equal be empowered. *Advances in Artificial Life*, pages 744–753.
- Klyubin, A., Polani, D., and Nehaniv, C. (2005b). Empowerment: A universal agent-centric measure of control. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 128–135. IEEE.
- Klyubin, A., Polani, D., and Nehaniv, C. (2008). Keep your options open: an information-based driving principle for sensorimotor systems. *PLoS ONE*, 3(12):e4018.
- Lungarella, M., Pegors, T., Bulwinkle, D., and Sporns, O. (2005). Methods for quantifying the informational structure of sensory and motor data. *Neuroinformatics*, 3(3):243–262.
- Pfeifer, R., Bongard, J., and Grand, S. (2007). *How the body shapes the way we think: a new view of intelligence*. The MIT Press.
- Prokopenko, M., Gerasimov, V., and Tanev, I. (2006). Evolving spatiotemporal coordination in a modular robotic system. *From Animals to Animats 9*, pages 558–569.
- Rasmussen, C. and Williams, C. (2006). *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, MA.
- Salge, C., Glackin, C., and Polani, D. (2012). Approximation of empowerment in the continuous domain. *Advances in Complex Systems*, 16(1 & 2):1250079.
- Schmidhuber, J. (1991). Curious model-building control systems. In *Neural Networks, 1991. 1991 IEEE International Joint Conference on*, pages 1458–1463 vol.2.
- Scott-Phillips, T. C., Dickins, T. E., and West, S. A. (2011). Evolutionary theory and the ultimate-proximate distinction in the human behavioral sciences. *Perspectives on Psychological Science*, 6(1):38–47.
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell Sys. Tech. Journal*, 27:623–656.
- Sims, K. (1994). Evolving 3d morphology and behavior by competition. *Artificial life*, 1(4):353–372.
- Steels, L. (2004). The autotelic principle. *Embodied Artificial Intelligence*, pages 629–629.
- Telatar, E. (1999). Capacity of multi-antenna gaussian channels. *European transactions on telecommunications*, 10(6):585–595.
- Touchette, H. and Lloyd, S. (2000). Information-theoretic limits of control. *Phys. Rev. Lett.*, 84:1156–1159.
- von Uexküll, J. (1909). *Umwelt und Innenwelt der Tiere*. Springer.

## Evolved digital ecosystems: Dynamic steady state, not optimal fixed point

Randal S. Olson<sup>1,2</sup>, Masoud Mirmomeni<sup>1,2</sup>, Tim Brom<sup>1,2</sup>, Eric Bruger<sup>1,3</sup>,  
Arend Hintze<sup>1,3</sup>, David B. Knoester<sup>1,3</sup>, and Christoph Adami<sup>1,3</sup>

<sup>1</sup>BEACON Center for the Study of Evolution in Action

<sup>2</sup>Department of Computer Science & Engineering

<sup>3</sup>Department of Microbiology & Molecular Genetics

Michigan State University, East Lansing, MI 48824

olsonran@msu.edu

### Abstract

Traditional models of ecosystems often assume that the species composing an unperturbed ecosystem become fixed so that only the relative abundances of the species change over time. Such ecosystems are said to have reached an optimal fixed point. However, recent work has suggested that neutral evolutionary processes can significantly alter the species composition of an ecosystem, allowing the ecosystem to exist in a dynamic steady state. Here, we investigate the stability of ecosystems and the nature of the equilibrium that forms using the digital evolution platform Avida, tracking evolving ecosystems over thousands of generations. We find that the communities that form are remarkably stable, and do not experience a significant loss of diversity in the long run even in experimental treatments where the communities suffer catastrophic population bottlenecks. When diversity rebounds, ecological communities are reconstituted in a different form than the one that was destroyed, but this difference is comparable to the difference the system would have accumulated if it had been left untouched. Thus, digital ecological communities exist in a dynamic steady state, which ultimately eliminates the effect of historical disturbances.

### Introduction

While the complexity of cellular and organismal biology is unquestionably stunning, it is often argued that the complexity of ecological communities is even more staggering, as they consist of co-adapted groups of organisms (Loehler, 2004). However, it is not immediately clear that ecological communities are necessarily any more complex. It is conceivable that general laws might guide the assembly, evolution, and even decay of ecosystems, simply because the interactions between species, as well as species with their environments, are simpler than the interactions between cellular components, or between cells within tissues that compose an organism. Indeed, simple ecosystems are usually modeled by systems of coupled differential equations that keep track of species and resource abundances (Tilman, 1982). In such models, ecosystems frequently exhibit an *ecological* steady state (Brock, 1967; Deakin, 1975; Aoki, 1988; Michaelian, 2005). In this state, resources flow through the system by being consumed and

replaced. Individuals come and go, but the species composition of the community is largely intact over large time scales. If this is so, then from the point of view of the species composition, the system has actually reached an *optimal fixed point*. In other words, the identity and frequency of a species is selected for, and does not change in the long run. Such ecological fixed points have been found experimentally in small systems (with a handful of species) (Rainey and Travisano, 1998) with evolution limited to only several weeks. Other experiments have found that communities will display different patterns of succession upon disruption by bottlenecks (e.g., in gut microbiota after administration of antibiotics), but the community ultimately arrives at a new stable state (Peterfreund et al., 2012).

It is difficult to ascertain whether any of these observations carry over to real ecological assemblies because tracking ecosystems over geological times is not possible, and modeling of such communities with standard methods such as systems of differential equations cannot shed light on this issue. While the stability of ecological communities can be studied (May, 1972, 1974; Montoya et al., 2006; Mougi and Kondoh, 2012), the existence of a dynamic steady state—where the community is constantly changing over evolutionary time scales and the only (approximate) constant is the number of species—cannot be studied because in the standard mathematical descriptions the number of possible participants is necessarily fixed from the outset. In contrast, in a dynamic steady-state, new species constantly emerge and established ones go extinct, while the ecological cohesion of the community remains intact.

If ecological assemblies are governed predominantly by neutral evolutionary processes (see, e.g., Chu and Adami 1999; Hubbell 2001; Volkov et al. 2003) rather than niche-specific adaptation, then dynamically changing fixed points should be expected. Here, we use digital evolution (Adami, 1998; Ofria and Wilke, 2004; Adami, 2006) as a tool to study the question of ecosystem evolution and stability from an “experimental” rather than mathematical point of view (see also Fortuna et al. 2013). We put the word experimental in quotes because not everyone is satisfied that what we learn

from digital experiments can carry over to biological assemblies of species. However, a significant amount of work with digital models has shown that they reproduce the basic phenomena associated with long-term evolution (Lenski et al., 2003; Wagenaar and Adami, 2004; Adami, 2006). Digital evolution experiments have even pointed to undiscovered effects in evolutionary theory (Wilke et al., 2001), which have subsequently been verified in “biochemicals.” The adaptive radiation of species in Avida has been studied previously (Cooper and Ofria, 2002; Chow et al., 2004; Walker and Ofria, 2013), but only a handful of studies have investigated the role of chance events on the outcome of evolution in digital systems. Previous studies on fluctuating environments, such as periods of resource scarcity (Yedid et al., 2008) and sudden changes in environment resource compositions (Wagenaar and Adami, 2004), and their effects on the evolution of specific tasks (i.e., specializing on a specific resource) have hinted that chance events do indeed affect the final outcome of evolution. Additionally, press (gradual) and pulse (instant) extinctions have been shown to alter the evolutionary path of a population enough to result in an entirely dissimilar final population (Yedid et al., 2009). Finally, an analysis of different forms of perturbations on digital ecosystems (such as mass extinctions) has shown that they affect the phylogenetic structure of the population, but leave little trace elsewhere (Yedid et al., 2012). These promising results highlight the need for more experiments studying the impact of historical contingency in the realm of digital evolution.

Here, we investigate the impact of population bottlenecks on the species composition of populations observed over the course of digital evolution. First, we show that population bottlenecks—even bottlenecks as small as a single organism—do not change the mean number of species in an ecosystem in the long run. Next, we provide evidence that populations evolve to use the same resources regardless of whether they experience a bottleneck. Finally, we demonstrate that while these populations use the same resources, the species that compose these populations do not remain at a single optimal fixed point. Rather, we suggest that evolving digital populations are in a dynamic steady state.

## Methods

We use the digital evolution platform Avida (Adami, 1998; Ofria and Wilke, 2004; Adami, 2006) to investigate the impact of population bottlenecks on populations of evolving digital organisms over long periods of evolutionary time. Avida has previously been used to investigate many fundamental aspects of evolution, including the evolutionary origins of complexity (Lenski et al., 2003), genetic organization (Misevic et al., 2006), adaptive radiation (Chow et al., 2004), and the division of labor (Goldsby et al., 2012). In this study, we subject the evolving populations to bottlenecks of varying size, then compare (1) the num-

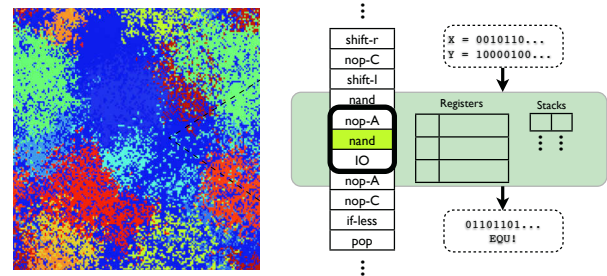


Figure 1: An Avida population containing multiple genomes (left) and the internal structure of an individual organism, called an Avidian (right).

ber of species, (2) the resource usage of the entire population, and (3) the resource usage of individual species between each experimental treatment. With these three measurements, we experimentally determine whether chance events such as population bottlenecks can significantly alter the evolutionary result of an evolving population. In the remainder of this section, we describe the main features of Avida and the experimental design of the study presented in this paper. All experiments were conducted with Avida version 2.12.3, which can be freely downloaded from <http://avida.devosoft.org/>.

## Avida

Figure 1 shows a typical Avida population and the internal structure of a digital organism, called an Avidian. These Avidians metabolize resources and reproduce in a common environment that is split up into individual cells, where a single Avidian inhabits each cell. During their lifetime, the Avidians execute their genome—a circular list of assembly-like instructions—using their virtual CPU. Executing these instructions allows the Avidians to perform various tasks in the environment (e.g., metabolize resources, described in more detail below), which can be thought of as the Avidian’s phenotype. In this study, each Avidian’s virtual CPU contains a circular list of three general-purpose registers, two general-purpose stacks, and four special-purpose heads, which are pointers into the Avidian’s genome, similar to a traditional program counter and stack pointer.

Further, each Avidian in this study is self-replicating, which means that it must contain instructions in its genome to copy itself and produce an offspring. During the self-replication process, the genome copy experiences mutations that change a single instruction to a different random instruction. Once the Avidian finishes copying itself, the copy is placed into a random cell elsewhere in the environment, i.e., the population is well-mixed. If the chosen cell is already inhabited by another Avidian, the existing Avidian is replaced by the new Avidian. By repeatedly following this metabolism-replication-mutation process, the Avid-

ian population is able to evolve and adapt to the environment over long time periods.

The Avida environment can be thought of as a “digital chemostat,” where simulated resources are constantly flowing in and out of the environment at predefined rates. Avidian genomes change over evolutionary time, and adapt to perform various logic tasks (e.g., AND, OR, and XOR), because the performance of such tasks is rewarded by “SIP” (single instruction processing) units. Each SIP unit gives an Avidian the ability to execute exactly one instruction, and can be thought of as the digital equivalent of ATPs, which power biochemical cells. Without SIPs, Avidian genomes cannot be executed. In order to perform a logic task, the Avidian program must have the correct sequence of instructions to input random binary numbers from the environment, perform a computation on them using a single logic instruction available to them (NAND), then write the resulting value back into the environment. At the same time, a resource associated with that logic task must be present in the environment. Because complex logic operations (such as EQU and XOR) can and must be built from simpler ones, Avidians must evolve the equivalent of metabolic pathways, only on a computational level. As an Avidian metabolizes more and more resources over its lifetime, it is able to execute more instructions faster than Avidians that have not metabolized any resources. Consequently, Avidians are indirectly selected to adapt to their environment and consume the available resources in the digital chemostat.

In this study, we use the “resource-9” environment, in which 9 logic tasks (NOT, NAND, AND, ORN, OR, ANDN, NOR, XOR, and EQU) are rewarded equally for completing them. The resource associated with each task flows into the digital chemostat at a fixed rate of 10 units/update. In general this rate can be varied, but we chose here the level at which the highest speciation rate was observed in Chow et al. (2004). Each Avidian can only consume each particular resource up to 5 times per update. Because resources are limited, the average amount of resource an Avidian consumes is proportional to the mean abundance of that resource across the population. In this limited resource environment, generalists that consume all 9 resources are selected against because they would consume each and every resource to the point that the net benefit of generalization is smaller than if each species specializes on one resource. As a consequence, mutants that evolve to tap into an unused resource have an advantage at first, and over time communities assemble that divide up the resource space roughly equally (as each resource is valued the same).

Any settings differing from the Avida defaults are described in Table 1. These settings are drawn from Chow et al. (2004) to replicate their Avida adaptive radiation experiments.

Setting	Value
Copy mutation rate	0.005
Insertion/deletion mutation rate	0.0
Min/max genome length	100
Max population	3000

Table 1: Custom Avida settings for this study.

## Control and bottleneck experiments

As a control, we first perform a set of Avida experiments for  $10^6$  updates with no population bottlenecks. These experiments provide a base expectation for what the evolved communities should look like if bottlenecks have no impact on the evolutionary outcome of a population. Next, we carry out another set of Avida experiments for  $10^6$  updates, but with the populations experiencing a single bottleneck of varying sizes (1, 5, 10, 20, 100, 200, 300, 400, and 500) at update  $5 \times 10^5$ . We execute the bottleneck procedure by removing random Avidians from the population until the population is reduced to the desired bottleneck size. After the bottleneck is applied, we allow the population to evolve without intervention for the remaining  $5 \times 10^5$  updates.

We initialize each Avida experiment with the same default ancestor, an Avidian with a genome length of 100 that is only capable of self-replicating. We repeat each experiment in replicate 100 times with random number seeds of 1-100. Before every bottleneck and at the end of every Avida run, we record the entire current population and the population history for use in a species clustering algorithm (in order to count species), described below. In addition, we collect the standard Avida statistics (averages, counts, resource, tasks, etc.) every 100 updates to perform population resource usage comparisons.

## Species clustering algorithm

To determine the species present in a population, we employ the species clustering algorithm from Chow et al. (2004), which clusters species based on phylogenetic distance. We calculated the phylogenetic distance between two Avidians by counting the number of ancestors between them along the lines of descent leading to their last most recent common ancestor. First, the algorithm requires the user to calibrate a threshold phylogeny depth value ( $T$ ) by calculating the  $T$  value necessary for the clustering algorithm to predict  $\leq 25\%$  of runs having 2 species,  $\leq 2.26\%$  of having 3 species, and  $\leq 0.1\%$  showing 4 species, when the algorithm is run on a set of 100 or more Avida runs with unlimited resources. It is known that when resources are unlimited, generalists will evolve, and the community will have exactly one species (Cooper and Ofria, 2002; Chow et al., 2004). With this calibrated  $T$  (here,  $T = 200,142$ ), the clustering algorithm then forms clusters of species in the reconstructed phylogeny by grouping genotypes less than  $T$  away from the



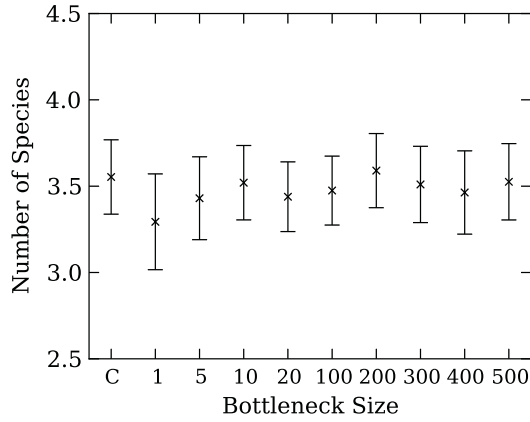


Figure 2: Average number of species for differing experiments based on the phylogenetic depth clustering algorithm. Each experimental treatment is listed along the bottom. The control experiment is labeled “C” and the bottleneck experiments are labeled with the size of the bottleneck. Error bars are two standard errors over 100 replicates.

computed genotype “species basins.” After the clustering algorithm identifies all of the species clusters, it outputs (1) the number of species and (2) the representative genotype of each species basin. This output allows us to compare species counts and species resource utilization between experimental treatments with and without population bottlenecks to determine whether the bottleneck had a significant effect on the evolutionary outcome of the population. The number of species predicted by this algorithm compares well with the “ecological” number of species, which is obtained by turning off mutation rate and counting the number of genotypes that remain in equilibrium after a long time (Cooper and Ofria, 2002; Chow et al., 2004).

### Difference in resource usage

After identifying the species for a given time point, we use Avida’s Analyze mode to determine each of the species’ resource utilization vector  $\vec{\phi} = (\phi_1, \phi_2, \dots, \phi_9)$ , where  $\phi_r$  is the average number of times the species has obtained resource  $r$  (associated with task  $r$ ) during its lifetime. We then normalize this vector so that the  $\phi_r$  of the resource that is used most by that species is set to 1.0.

In order to calculate the difference in resource usage between two species  $i$  and  $j$ , consider two resource utilization vectors  $\vec{\phi}_i$  and  $\vec{\phi}_j$ . We define the difference in utilization between those species as the Euclidean distance  $d_{ij} = |\vec{\phi}_i - \vec{\phi}_j|$ . What is the difference between two communities? If community  $C_a$  is defined by the assembly  $C_a = (\vec{\phi}_1, \dots, \vec{\phi}_n)$  and community  $b$  by  $C_b = (\vec{\phi}_1, \dots, \vec{\phi}_m)$ , we first pad the assembly vector  $C$  of the community with the smaller number of species with null vectors, and define

the assembly difference matrix as

$$D_{ij}^{(ab)} = |\vec{\phi}_i^{(a)} - \vec{\phi}_j^{(b)}|, \quad (i, j = 1, \dots, n). \quad (1)$$

Because this distance depends on the ordering of species in the community vector, we define the community distance  $D$  as the minimum of the trace of the distance matrix, minimized over all permutations of the species order. Thus, let  $P$  be a permutation matrix (of the set of  $n!$ ). Then

$$D = \min_P \text{Tr}(PD^{(ab)}). \quad (2)$$

In other words, to find the difference between two communities, we compute all pairwise distances between the species of both populations. If both communities are identical, the sum of the diagonal of this pairwise distance matrix must be 0.0, but only if we have correctly matched all species. If the populations have a different number of species, we supplement the population with fewer species with a species using no resources. To perform the match, we test all permutations of the distance matrix (i.e., with different species orders) to minimize the trace (the sum of the diagonal elements) of the matrix. This measure provides the minimum distance between two communities in species resource usage space.

## Results

### Species counts

Figure 2 shows the species counts based on phylogenetic depth for the control experiment in comparison to the varying bottleneck experiments. On average, the control resulted in  $3.55 \pm 0.22$  species (mean  $\pm$  two standard errors) and none of the experiments resulted in a significantly different species count. It is interesting to note that even experiments with a bottleneck size of only one organism did not have their ultimate species counts significantly impacted.

### Comparison of task distributions

Next, we compare the average population resource usage

$$\vec{R} = \frac{1}{N_{\text{tot}}} (N_1, \dots, N_9), \quad (3)$$

where  $N_r$  is the number of times resource  $r$  has been consumed by the population *per update*, and  $N_{\text{tot}} = \sum_{i=1}^9 N_i$ , for the final populations of each experiment. Differences in  $\vec{R}$  allow us to examine if there is a significant difference in overall resource usage before and after bottlenecks of different sizes.

Figure 3 shows the  $\vec{R}$  of the final control and bottleneck populations. Qualitatively, there appears to be little difference in the resource usage between the different experiments, indicating that the populations recovered from the bottleneck and eventually reconstituted an ecosystem that consumes resources at a rate comparable to an untouched



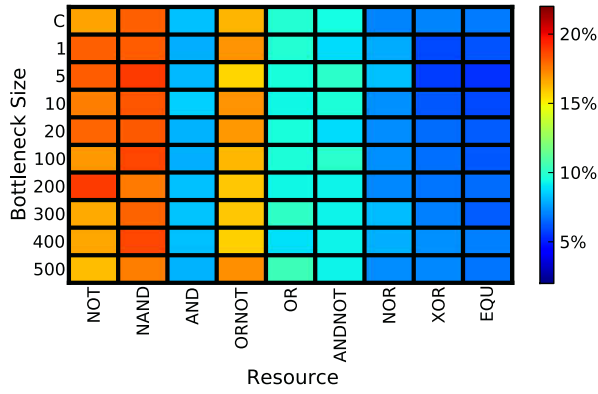


Figure 3: Average fraction of tasks performed per update  $R_r$  (defined in Equation 3) by the Avidians in the final population of different experiments over 100 replicates. The experiments are listed along the left side. The control population at update  $10^6$  is labeled “C” and the experimental populations are labeled with the size of the bottleneck. Each task along the bottom is a logical function in Avida which can be considered a resource that a digital organism can adapt to metabolize.

ecosystem. To confirm our qualitative analysis, we compute the Pearson correlation coefficient between the  $\vec{R}$  of the bottleneck populations and the control populations. The smallest correlation is between the control populations and the 5-organism bottleneck experimental populations ( $\rho = 0.98$ ), which still indicates a strong correlation in the resource usage vectors. Thus, even the most severely bottlenecked populations reconstituted the same resource usage after a long period of evolutionary time, even though the species composition could be very different. Similar overall resource usage by different communities could be an indication of *functional redundancy* (Tilman et al., 1997; Wohl et al., 2004).

### Comparison of individual species resource usage

The populations evolve to use the same resources over all experimental conditions, but the species within a population (the community assembly) may look very different from one experiment to another. To establish a baseline, we look at the differences in species resource usage among the experimental populations after  $5 \times 10^5$  updates. Shown in Figure 4, we compute the mean difference in species resource usage between communities evolved in 100 independent populations (excluding a direct comparison of a population with itself) and find  $D = 2.26 \pm 0.025$ . With this measure, we characterize the differences that arise in communities simply because each population takes its own historical path.

Next, we compare the communities between the reference populations at  $5 \times 10^5$  updates to two sets of populations at update  $10^6$ : One set of control populations that never experienced a population bottleneck, and another set of experimen-

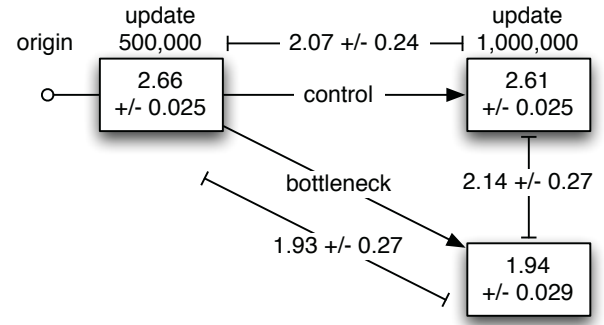


Figure 4: Overview of the population's species resource usage differences. The values shown are the mean difference  $D$  ( $\pm$  two standard errors) between and within populations. All populations from update 0 (labeled “origin”) to update  $5 \times 10^5$  had the same evolutionary history. At update  $5 \times 10^5$ , the experimental populations (labeled “bottleneck”) experienced a single bottleneck reducing the population to one organism, whereas the control populations (labeled “control”) were untouched. After the treatment at update  $5 \times 10^5$ , the populations were then allowed to evolve for another  $5 \times 10^5$  updates. The resulting populations are labeled “update 1,000,000.”

tal populations that experienced a severe population bottleneck (a single organism) at update  $5 \times 10^5$ . We found that there was no significant difference in inter-population differences between the reference populations at update  $5 \times 10^5$  (mean  $\pm$  two standard errors,  $D = 2.66 \pm 0.025$ ) and the control population at update  $10^6$  ( $D = 2.61 \pm 0.025$ ). In contrast, the inter-population differences within the experimental populations were significantly reduced ( $D = 1.94 \pm 0.029$ ).

While populations evolve to use the same resources regardless of treatment (Figure 3), it is not clear whether or not the populations are at a dynamic steady state or an optimal fixed point. If the populations do not change over evolutionary time (i.e., the populations are at an optimal fixed point), we would expect the difference in species resource usage between the baseline populations at update  $5 \times 10^5$  and the control populations at update  $10^6$  to be minimal, if not 0.0. Instead, when comparing each control population at update  $10^6$  with its corresponding reference population at update  $5 \times 10^5$ , we observe that the populations are composed of significantly different species (Figure 4,  $D = 2.07 \pm 0.24$ ). Additionally, we find a significant difference when performing the same comparisons between the reference populations and experimental populations that experienced a population bottleneck ( $D = 1.93 \pm 0.27$ ). This is the same difference that we find when we again perform the same comparison between the control populations and experimental populations at update  $10^6$  ( $D = 2.14 \pm 0.27D$ ). Thus, although

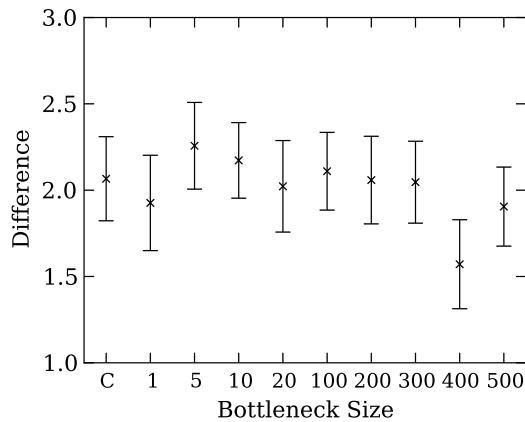


Figure 5: Mean difference  $D$  in species resource usage between the populations at update  $5 \times 10^5$  and the populations at update  $10^6$ . Each experimental treatment is listed along the bottom. The control populations are labeled “C,” whereas the bottleneck populations are labeled with the size of the bottleneck. Error bars are two standard errors over 100 replicates.

the experimental populations are significantly different from the reference populations, they are just as different as they would have become if they never experienced a population bottleneck. Together, these data highlight our two major findings:

- (1) population bottlenecks do not have a significant effect on the species composition of a population over long evolutionary periods, and
- (2) over sufficiently long evolutionary periods, populations are in a dynamic steady state rather than at an optimal fixed point.

In a catastrophic population bottleneck, only one organism survives the bottleneck, which effectively destroys the ecosystem and reduces the number of species to 1. By subjecting the populations to such a severe population bottleneck, the population is forced to re-evolve every other species, which may explain the results above. What if the populations experience a less severe population bottleneck? A less severe population bottleneck would preserve most, if not all, of the ecosystem and its species. In Figure 5, we further demonstrate that regardless of the population bottleneck size, populations do not maintain an optimal fixed point. Additionally, we show in Figure 6 that regardless of the population bottleneck size, all experimental populations at update  $10^6$  have the same difference from the control population at update  $10^6$ . Thus population bottleneck size does not affect the species composition of populations over sufficiently long evolutionary periods.

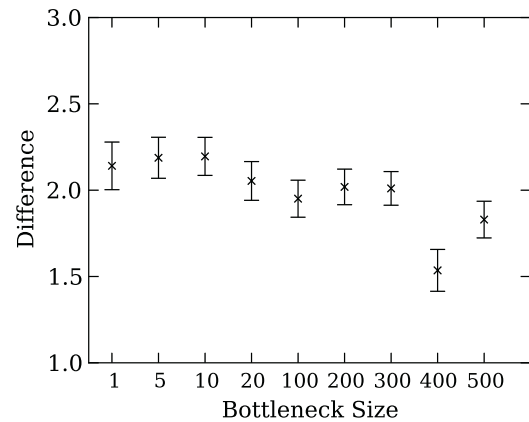


Figure 6: Mean difference  $D$  in species resource usage between the control populations at update  $10^6$  and the bottleneck populations at update  $10^6$ . Each experimental treatment is listed along the bottom. The bottleneck populations are labeled with the size of the bottleneck. Error bars are two standard errors over 100 replicates.

## Discussion

Competition over resources shapes ecological communities, and creates assemblies that are highly adapted to their environment. Species (or ecotypes in microbial communities) can only be maintained if they are adapted to different niches, which means that they must each “make a living” differently. In our model system, this means that each species must specialize to predominantly use a different resource. Here we have asked: Once an ecosystem is established, will it maintain its species composition over long periods of time (i.e., an optimal fixed point), or do species continue to change over evolutionary time (i.e., a dynamic steady-state)?

We find that populations evolve the same number of species regardless of the bottleneck size, and that the number of species in a population is much smaller than the number of available resources (on average around 4, compared to the theoretical maximum of 9). Each population has approximately the same distribution of consumed resources, again regardless of experimental conditions. Analyzing populations in detail, we find that species partition the resources (i.e., niches) in many different ways, and continue to do so during evolution. While technically speaking, no new species form after the establishment of a community (as opposed to what is observed in perfectly neutral models of species diversity, e.g. de Aguiar et al. (2009), where the rate of speciation is constant over time), we notice that the species themselves continue to change, and the community with them. Applying bottlenecks of different sizes, including catastrophic events where only a single organism survives, has no effect on this phenomenon. Ecosystems re-

form after the catastrophic event (either in a form similar to the community before the event, or differently), but continue to change thereafter. Thus, evolving ecosystems resemble a dynamic steady-state rather than an optimal fixed point.

These results have significant implications for experimentalists who work with biological systems that require regular bottlenecks on the population to conduct the experiment, e.g., the *E. coli* long term evolution experiment (LTEE) (Lenski, 2011). This study demonstrates that these regular population bottlenecks do not affect the long-term evolution of populations, nor do they significantly affect the species composition of the population in the long-term. We note that at least one population of the LTEE seems to have developed a community of coexisting types (Blount et al., 2012).

Prior biological experiments suggested that population bottlenecks imposed on ecological communities leads to several waves of succession followed by the establishment of a new stable state with a similar degree of diversity compared to the initial stable state (Peterfreund et al., 2012). These experiments, however, were all conducted on a very short time scale. We evolved our populations for 25,000 generations between measurements (assessment of species composition), which allows for much more neutral evolution. We note that in these experiments, each of the 9 possible resources were worth the same to an Avidian, i.e., switching from one resource to another would not be beneficial (nor detrimental) as long as the concentration of that resource in the community is the same. It is possible that this setting creates more neutrality in the landscape compared to a setting where each resource has a distinct metabolic payoff, and it would be interesting to study a fitness landscape with different metabolic payoffs in detail in future work.

It might also seem surprising that we observe drift in the community even though the number of species in the community is quite low (between 2-6, on average). Most of the interesting biological communities consist of many more species: It has even been suggested that soil microbial communities could harbor up to  $10^6$  species (Gans et al., 2005). It would be interesting to test community drift and turnover when there are an order of magnitude more niches to be occupied, which can be done in Avida by placing digital organisms in the “logic-77” environment, giving 77 distinct niches. We have also not addressed the effect of trophic levels on ecosystem stability and turnover. Recent modeling efforts (Mougi and Kondoh, 2012) suggest that the variety of trophic interactions stabilize these communities, which could in principle lead to a reduction in community drift.

## Conclusions

We found that populations of digital organisms exposed to an environment with limited resources rapidly radiate to take advantage of the available niches, but that the rate of speciation stops long before all niches are occupied. Severe bottle-

necks can destroy these communities, but stable communities rapidly re-evolve, albeit with a different species composition. We have shown that the species composition of these communities is not affected by bottlenecks of any size in the long run, simply because these communities are in a state of constant flux anyway: The communities form dynamic steady-states, where the species are constantly changing the resources they specialize on. While the evolved communities are resistant to invasion (Chow et al., 2004), they are not resistant to change. Because the available niches can be occupied by a multitude of functionally similar or even identical species (and perhaps because each resource in the logic-9 environment is worth the same), the communities themselves are subject to a considerable amount of drift, even when the community as a whole remains cohesive. The communities are resistant to invasion due to the particular trade-offs each species has incurred in its adaptive specialization. In this respect, Avidian communities behave much as predicted by Tilman’s “stochastic niche theory” (Tilman, 2004): They are dominated by both adaptive forces (generating the trade-offs) as well as neutral forces (stochastic assembly and drift). Thus, we suggest that further experimentation with Avidian ecosystems can generate significant progress in our understanding of ecological theory and experiments.

## Acknowledgements

Part of this work (including all data) is the result of a BEACON class final project (CSE 845) at Michigan State University. We would like to thank the instructors Drs. Charles Ofria and Ian Dworkin for their advice on the interpretation of the results during the class period. This work was supported in part by the Paul G. Allen Family Foundation, the National Science Foundation (NSF) BEACON Center for the Study of Evolution in Action under Cooperative Agreement DBI-0939454, and NSF grant OCI-1122617. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF. We wish to acknowledge the support of the Michigan State University High Performance Computing Center and the Institute for Cyber Enabled Research (iCER).

## References

- Adami, C. (1998). *Introduction to Artificial Life*. TELOS Springer Verlag, New York, NY.
- Adami, C. (2006). Digital genetics: unravelling the genetic basis of evolution. *Nat Rev Genet*, 7(2):109–118.
- Aoki, I. (1988). Entropy laws in ecological networks at steady state. *Ecological Modelling*, 42:289–303.
- Blount, Z. D., Barrick, J. E., Davidson, C. J., and Lenski, R. E. (2012). Genomic analysis of a key innovation in an experimental *escherichia coli* population. *Nature*, 489:513–518.

- Brock, T. D. (1967). The ecosystem and the steady state. *BioScience*, 17:166–169.
- Chow, S. S., Wilke, C. O., Ofria, C., Lenski, R. E., and Adami, C. (2004). Adaptive radiation from resource competition in digital organisms. *Science*, 305(5680):84–6.
- Chu, J. and Adami, C. (1999). A simple explanation for taxon abundance patterns. *Proc Natl Acad Sci U S A*, 96(26):15017–9.
- Cooper, T. and Ofria, C. (2002). Evolution of stable ecosystems in populations of digital organisms. In Standish, R. K., Bedau, M. A., and Abbass, H. A., editors, *Proceedings of the Eighth International Conference on Artificial Life*, pages 227–232.
- de Aguiar, M. A. M., Baranger, M., Baptestini, E. M., Kaufman, L., and Bar-Yam, Y. (2009). Global patterns of speciation and diversity. *Nature*, 460(7253):384–7.
- Deakin, M. A. B. (1975). The steady states of ecosystems. *Mathematical Biosciences*, 24:319–331.
- Fortuna, M. A., Zaman, L., Wagner, A. P., and Ofria, C. (2013). Evolving digital ecological networks. *PLoS Comput Biol*, 9(3):e1002928.
- Gans, J., Wolinsky, M., and Dunbar, J. (2005). Computational improvements reveal great bacterial diversity and high metal toxicity in soil. *Science*, 309(5739):1387–90.
- Goldsby, H. J., Dornhaus, A., Kerr, B., and Ofria, C. (2012). Task-switching costs promote the evolution of division of labor and shifts in individuality. *Proc Natl Acad Sci U S A*, 109(34):13686–13691.
- Hubbell, S. P. (2001). *The Unified Neutral Theory of Biodiversity and Biogeography*. Princeton University Press, Princeton, NJ.
- Lenski, R. E. (2011). Evolution in action: a 50,000 generation salute to charles darwin. *Microbe*, 5:30–33.
- Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423(6936):139–144.
- Loehler, C. (2004). Challenges of ecological complexity. *Ecological Complexity*, 1:3–6.
- May, R. M. (1972). Will a large complex system be stable? *Nature*, 238:413–414.
- May, R. M. (1974). *Stability and Complexity in Model Ecosystems*. Princeton University Press, Princeton, N.J.
- Michaelian, K. (2005). Thermodynamic stability of ecosystems. *J. Theor. Biol.*, 237:323–335.
- Misevic, D., Ofria, C., and Lenski, R. E. (2006). Sexual reproduction reshapes the genetic architecture of digital organisms. *Proc Biol Sci*, 273(1585):457–64.
- Montoya, J. M., Pimm, S. L., and Solé, R. V. (2006). Ecological networks and their fragility. *Nature*, 442(7100):259–64.
- Mougi, A. and Kondoh, M. (2012). Diversity of interaction types and ecological community stability. *Science*, 337(6092):349–51.
- Ofria, C. and Wilke, C. O. (2004). Avida: a software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229.
- Peterfreund, G. L., Vandivier, L. E., Sinha, R., Marozsan, A. J., Olson, W. C., Zhu, J., and Bushman, F. D. (2012). Succession in the gut microbiome following antibiotic and antibody therapies for *Clostridium difficile*. *PLoS One*, 7(10):e46966.
- Rainey, P. B. and Travisano, M. (1998). Adaptive radiation in a heterogeneous environment. *Nature*, 394:69–72.
- Tilman, D. (2004). Niche tradeoffs, neutrality, and community structure: a stochastic theory of resource competition, invasion, and community assembly. *Proc Natl Acad Sci U S A*, 101(30):10854–61.
- Tilman, D., Knops, J., Wedin, D., Reich, P., Ritchie, M., and Siemann, E. (1997). The influence of functional diversity and composition on ecosystem processes. *Science*, 277:1300–1302.
- Tilman, G. D. (1982). *Resource Competition and Community Structure*. Princeton University Press, Princeton, N.J.
- Volkov, I., Banavar, J. R., Hubbell, S. P., and Maritan, A. (2003). Neutral theory and relative species abundance in ecology. *Nature*, 424(6952):1035–7.
- Wagenaar, D. A. and Adami, C. (2004). Influence of Chance, History, and Adaptation on Digital Evolution. *Artificial Life*, 10(2):181–190.
- Walker, B. L. and Ofria, C. (2013). Evolutionary potential is maximized at intermediate diversity levels. In Adami, C., Bryson, D. M., Ofria, C., and Pennock, R. T., editors, *Proceedings 13th International Conference on the Simulation and Synthesis of Life*, pages 116–120.
- Wilke, C. O., Wang, J. L., Ofria, C., Lenski, R. E., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412(6844):331–333.
- Wohl, D. L., Arora, S., and Gladstone, J. R. (2004). Functional redundancy supports biodiversity and ecosystem function in a closed and constant environment. *Ecology*, 85:1534–1540.
- Yedid, G., Ofria, C. A., and Lenski, R. E. (2008). Historical and Contingent Factors Affect Re-evolution of a Complex Feature Lost During Mass Extinction in Communities of Digital Organisms. *Journal of Evolutionary Biology*, 21(5):1335–1357.
- Yedid, G., Ofria, C. A., and Lenski, R. E. (2009). Selective press extinctions, but not random pulse extinctions, cause delayed ecological recovery in communities of digital organisms. *The American Naturalist*, 173(4):E139–E154.
- Yedid, G., Stredwick, J., Ofria, C. A., and Agapow, P.-M. (2012). A comparison of the effects of random and selective mass extinctions on erosion of evolutionary history in communities of digital organisms. *PLoS One*, 7(5):e37233.



# Prebiotic Evolution of Molecular Assemblies: From Molecules to Ecology

Markovitch Omer<sup>1</sup> and Lancet Doron<sup>1</sup>

<sup>1</sup>Department of Molecular Genetics, Weizmann Institute of Science, Rehovot 76100, Israel  
omermar@gmail.com

## Abstract

Present life portrays a two-tier phenomenology: molecules compose supramolecular structures, such as cells or organisms, which in turn portray population behaviors, including selection, evolution and ecological dynamics. Prebiotic models have often focused on evolution in populations of self-replicating supramolecules, without explicitly invoking the intermediate molecular-to-supramolecular stage. We explore a prebiotic model that allows one to relate parameters of chemical interaction networks within molecular assemblies to emergent ecological and evolutionary properties in populations of such assemblies. We use the graded autocatalysis replication domain (GARD) model, which simulates the network dynamics of amphiphile-containing molecular assemblies, and exhibits quasi-stationary compositional states termed *compotypes*. These grow by catalyzed accretion, divide and propagate their compositional information to progeny in a replication-like manner. The model allows us to ask how molecular network parameters influence assembly evolution and population ecology, analyzable by a multi species logistic (r-K) model for population ecology (Lotka-Volterra competition model). We found that *compotypes* with a larger intrinsic molecular repertoire show a higher intrinsic growth (r) and lower carrying capacity (K), as well as lower replication fidelity. This supports a prebiotic scenario initiated by fast-replicating assemblies with a high molecular diversity, evolving into more faithful replicators with narrower molecular repertoires. A main difference from classical ecology is that in GARD species inter convert into each other rather than consume each other or compete on resources, thus representing ‘fast forward’ of speciation.

## Introduction

The path from organic mixtures (i.e., the primeval soup) to reproducing life-like protocells no doubt required the emergence of replicating systems capable of undergoing Darwinian evolution. Therefore, uncovering how such entities emerged in early niches will greatly contribute to our understanding of life’s origin and can potentially allow one to design novel experiments. The GARD model (Segre, Ben-Eli, Lancet 2000) in the realm of the lipid world scenario (Segre, Ben-Eli, Deamer et al. 2001) offers one possible route for such pursuit. In this framework, non-covalent assemblies of amphiphiles, such as lipid micelles or vesicles are studied. These store information in the form of nonrandom molecular compositions, which is passed to progeny via homeostatic growth accompanied by fission. The model quantitatively describes the details of such a process (Segre et al. 2000). It is based on a directed catalytic network (termed  $\beta$ ), whose nodes and edges respectively represent molecular types and catalytic

rate enhancements. Importantly, the system is kept away from thermodynamic equilibrium by assembly fission, which produces two progeny assemblies. Key in GARD dynamics are *composomes*, replication-prone quasi-stationary states. A group of *composomes*, gleaned by clustering, is termed *compotype*, and may be regarded as species in the framework of lipid world and GARD. Indeed, such GARD species were recently shown to display a significant measure of Darwinian evolution (Markovitch and Lancet 2012), in disagreement with a report (Vasas, Szathmáry, Santos 2010) criticizing this notion on the basis of testing random compositions and with no statistical rigor.

## Simulations

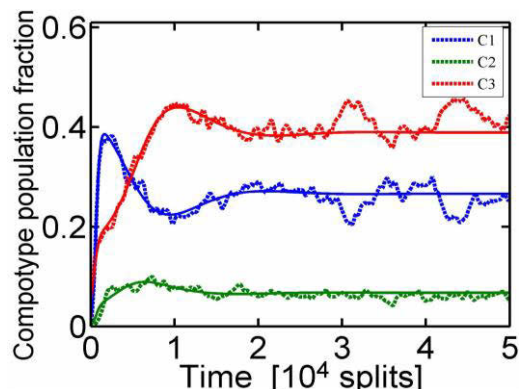
The GARD10 MATLAB code was employed for all simulations, using parameter values identical to those employed previously (Markovitch and Lancet 2012). The dynamics of compositional assemblies in a reactor under constant population conditions were examined. The reactor is seeded with 1,000 random compositions which are allowed to simultaneously grow based on their idiosyncratic kinetic parameters, and undergo fission when reaching a predefined maximal size. The pre-fission composition of each assembly is assessed as belonging to one of the *compotypes* characterizing the specific  $\beta$  or to “drift”.  $C_i$  marks the fractional number of assemblies belonging to *compotype*  $i$  (out of 1,000). Each simulation is performed for 50,000 split events in the reactor, typically sufficient to reach steady state in *compotype* frequencies. For statistical rigor, 1,000 such simulations were performed, each with a different  $\beta$  whose edges are randomly drawn from a lognormal distribution (Segre, Shenhav, Kafri et al. 2001).

## GARD Population Dynamics

Different simulations showed widely different dynamic behaviors, such as non-trivial “takeover” of a fast-rising *compotype* by a slower one (Fig. 1). Such dynamics is typical of natural ecosystems that harbor multiple species with competition or predator-prey relationships. The results were analyzed by a multi species logistic equation (May 1974; Gabriel, Saucy, Bersier 2005) (Fig. 1 legend). The GARD model thus affords a unique opportunity to directly relate molecular parameters to ecological behavior, bypassing the organismal complexity that usually bridges the two. As an example, the relationship between a *compotype*’s molecular



diversity and two central quantitative ecological parameters are portrayed here.



**Fig. 1:** An example of population dynamics. Broken lines represent the three compotype species found in this simulation and solid lines are a fit to the logistic growth:  $dC_i/dt = r_i C_i [K_i - C_i - \sum_j (\alpha_{ij} C_j)] / K_i$ , where  $C_i$  is the population-fraction of compotype  $i$  in the population at time  $t$ . Fitted parameter values are:  $r_{1,3} = 5e-3, 4e-3, 2e-3$ ;  $K_{1,3} = 0.56, 0.49, 0.72$ ;  $C(t=0)_{1,3} = 0.019, 0.003, 0.055$ ;  $\alpha_{12}=1.5, \alpha_{13}=0.05, \alpha_{21}=0.8, \alpha_{23}=0.56, \alpha_{31}=1.3, \alpha_{32}=0$ .

### Compotype Molecular Diversity

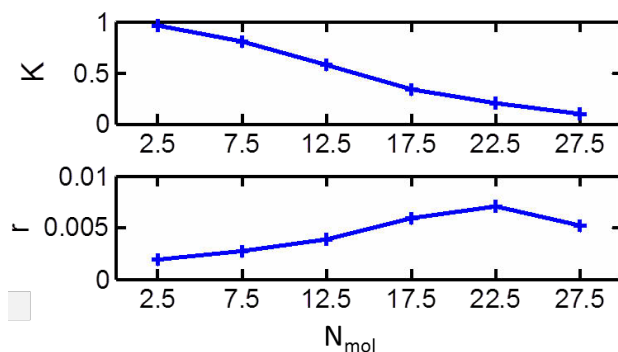
The chemistry to ecology transition was done by analyzing populations of GARD assemblies through the scope of multi species logistic growth. In this analysis, each compotype species  $i$  is characterized by two basic parameters: the intrinsic growth rate ( $r_i$ ) and the carrying capacity ( $K_i$ ).

The simulations result in the generation of about 1,500 compotypes in 1,000 chemical niches ( $\beta$  networks). Compotypes are defined at their molecular level by  $N_{mol}$ , the size of the intrinsic molecular repertoire of the compotype. In a simulation with  $N_G$  molecular types,  $N_{mol} < N_G$  represents the subset of molecule types present as a result of the intermolecular catalytic interactions in  $\beta$ . It is found that  $K$  values are inversely correlated with  $N_{mol}$ . In contrast,  $r$  values show a weak positive correlation (Fig. 2). Thus, in the absence of competition, the time-dependent prevalence of compotypes with large  $N_{mol}$  will show a steep ascent with a relatively low plateau, while those with low  $N_{mol}$  will show a slower ascent but can potentially reach a higher plateau.

Of note, an analog trend to the increase of the intrinsic growth rate with molecular diversity was observed in experimental data for 113 Bacteria (Freilich, Kreimer, Borenstein et al. 2009), whereby a negative correlation between measured doubling time and metabolic network size was found. The results might advocate for a prebiotic scenario initiated by fast-replicating assemblies with a high molecular diversity, evolving into more faithful replicators with narrower molecular repertoires. This is not unlike the transition from prebiotic “random chemistry” to the relatively restricted repertoire of small molecules (monomers) seen in present-day living cells (Segre et al. 2000).

### Compotype Replication Fidelity

In the absence of competition, i.e. when examining simulations exhibiting only one compotype species, a



**Fig. 2:** Compotypes dependence of carrying capacity ( $K$ ) and intrinsic growth rate ( $r$ ) on the size of the intrinsic molecular repertoire of compotypes ( $N_{mol}$ ). Data is binned.

compotypes'  $K$  values typically does not reach the upper limit of 1.0 (mean  $K=0.54 \pm 0.35$  for such simulations).  $K$  represents the maximal number of individuals that may be sustained in an environmental niche. In the original Verhulst formalism, death was introduced by as a potential solution to the Malthusian exponential growth, and later in the r-K formalism  $K$ =birth/death (Gabriel et al. 2005). In GARD, a similar interpretation of  $K$  pertains, whereby a positive correlation between  $K$  and replication fidelity ( $F_{rep}$ ) is observed:  $K=6.2 \cdot F_{rep}-5.35$  with  $R^2=0.63$ .  $F_{rep}$  measures the average degree of compositional similarity between a compotype assembly to its fully grown progeny. Thus, unfaithful replication means that the fully grown progeny has lost its compotype state and is considered drift, somewhat comparable to death.

### Acknowledgements

We thank Paul Higgs, Raphael Zidovezki and Natalio Krasnogor for discussions. This work is partly supported by EU-FP7 project MATCHIT and by the Crown Human Genome Center at the Weizmann Institute of Science.

### References

- Freilich S, Kreimer A et al. (2009) Metabolic-network-driven analysis of bacterial ecological strategies. *Genome Biol* 10 (6).
- Gabriel JR, Saucy F et al. (2005) Paradoxes in the logistic equation? *Ecol Model* 185 (1):147-151.
- Markovitch O, Lancet D (2012) Excess Mutual Catalysis Is Required for Effective Evolvability. *Artif Life* 18 (3):243-266.
- May RM (1974) Biological Populations with Nonoverlapping Generations - Stable Points, Stable Cycles, and Chaos. *Science* 186 (4164):645-647.
- Segre D, Ben-Eli D et al. (2001) The lipid world. *Origins of Life and Evolution of the Biosphere* 31 (1-2):119-145.
- Segre D, Ben-Eli D et al. (2000) Compositional genomes: Prebiotic information transfer in mutually catalytic noncovalent assemblies. *P Natl Acad Sci USA* 97 (8):4112-4117.
- Segre D, Shenhav B et al. (2001) The molecular roots of compositional inheritance. *Journal of Theoretical Biology* 213 (3):481-491.
- Vasas V, Szathmáry E et al. (2010) Lack of evolvability in self-sustaining autocatalytic networks constraints metabolism-first scenarios for the origin of life. *P Natl Acad Sci USA* 107 (4):1470-1475.

# Human-Robot Analogy – How Physiology Shapes Human and Robot Motion

Veljko Potkonjak<sup>1</sup>, Vladimir M. Petrović<sup>1</sup>, Kosta Jovanović<sup>1</sup> and Dragan Kostić<sup>2</sup>

<sup>1</sup>School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11000 Belgrade, Serbia

<sup>2</sup>Department of Mechanical Engineering, Technical University of Eindhoven, P.O. Box 513, 5600 MB Eindhoven, Netherlands  
potkonjak@yahoo.com

## Abstract

The fast grow of expectations from robots and the technical obstacles the robot developers face when trying to meet the requests, force an orientation towards designing and controlling robots by following biological paragons. This tendency increases interest in human-robot analogy, and the present work is a part of this stream. The paper questions the voluntariness of human motion by relating it to physiological processes. We concentrate on the important question of redundancy. One notes that humans do not resolve the redundancy on the level of consciousness, except in some specific examples (like obstacle avoidance), but rather on a lower level of decision-making – the human reaction to the faced problem is somehow "automatic". We suggest that this automatism is closely related to physiological processes, particularly to the progress of fatigue. Then we try to mathematically model these processes and their influence to human motion. The mathematical model of fatigue progress is derived as well as an algorithm for human-like redundancy resolution. We finally consider the implications of the obtained results to anthropomimetic robotics. The concept is verified by simulating the system behavior and comparing it qualitatively with the behavior of a human control group.

## Introduction

In the last decade we witness the fast growing interest in technical systems, in particular robots, mimicking living beings. A new class of robots have appeared – anthropomimetic robots – imitating humans regarding their mechanical structure, actuation, and intelligence (Holland & Knight, 2006; Diamond, et al., 2012; Potkonjak, et al., 2011; Wittmeier, et al. 2013; Mizuuchi, et al., 2007; Sodeyama, et al., 2008). Mimicking humans is an ultimate response to the increasing expectations and complexity of tasks imposed to robots, on one hand, and the technical and technological obstacles robot developers face when trying to meet the requirements, on the other.

With such approach the deep relations between motion/actuation, intelligence, and physiological processes have to be explored. For instance, motion used to be considered as a purely voluntary action resulting from consciousness and intelligent process. This was specially the case with robots where the voluntariness of motion was almost an axiom. However, the new concepts question this viewpoint. Regarding consciousness, it has been found that with humans a motor activity starts even before we are aware of the intention (Haggard & Libet, 2001). Researchers have also recognized the significance of morphology and motor

activities in shaping human intelligence (Pfeifer & Bongard, 2007). The present work does not elaborate these interesting problems but questions the voluntariness of human motion from another standpoint being dependant on the physiological processes. We concentrate on the important question of redundancy. The kinematic redundancy is normally present in humans and accordingly in anthropomimetic robots as well, offering multiple choice when deciding about joint motions that will execute some given end-effector motion task (e.g., some manipulation in space). One notes that humans do not resolve the redundancy on the level of consciousness, except in some specific examples (like obstacle avoidance), but rather on a lower level of decision-making – the human reaction to the faced problem is somehow "automatic". We suggest that this automatism follows from physiological processes, particularly from the progress of fatigue. Then we try to mathematically model these processes and their influence to human motion, to finally see whether the obtained results have sense and implications in anthropomimetic robotics. Human-like reactions of robot to overloading and a kind of human-like communication have been achieved.

## Background Research and the New Concept

### Kinematic Redundancy

From a mechanical point of view, a human and a robot resembling a human are kinematically redundant, i.e., their mechanisms feature a higher degree of mobility than required for a given motion in operational space. Kinematic redundancy contributes to motion dexterity and facilitates coping with unpredictable changes within its environment. Some advantages resulting from redundancy are exploited on the level of consciousness and intelligence (e.g., avoiding obstacles in the workspace) while others are exploited automatically, on a lower level of decision-making (this being the case with avoiding singularities and avoiding mechanical limits in joints). However, very often redundancy is not an advantage but rather a problem that needs resolution. Its implications are particularly emphasized in the well-known inverse kinematics (IK) problem. This is a problem of searching for joint motions that provide a desired trajectory of the end-effector in operational space. The presence of kinematic redundancy means that the same end-effector's trajectory can be executed with different joint motions. Hence,

the problem of the particular choice between available joint motions arises. If there are no specific constraints (like obstacles in workspace) that require the engagement of redundancy, then we have a "useless" surplus of joints and need to find the optimization criterion that will allow for the unique choice. The work done by Potkonjak et al. (2003) explored a typically human redundant task – handwriting, and showed little difference in results obtained by using engineering and biologically-inspired criteria for optimization.

Among optimization criteria which are mainly engineering, we mention the following. Joint movement time is an example of a kinematic cost function. Examples of dynamic cost functions are: quadratic norm of joint control torques (Hollerbach & Suh, 1987), kinetic energy (Khatib, 1983), jerks in joints (Hogan, 1984). Several neuro-physiological and psychophysical cost functions were also suggested (Sief-Naraghi & Winters, 1989): "input energy" was defined as a quadratic norm of input neural signals of motor units (muscles), while "input fatigue" denote the magnitude of such neural signals. The authors suggested some proper combination of these functions, rather than their separate application.

In this study, special attention is paid to functions of joint "discomfort", which were experimentally derived to identify arm postures of maximum comfort (Cruse, et al., 1990). They were determined upon analysis of recorded electromyographic (EMG) signals taken from subjects engaged in experiments, as well as by using their subjective psychophysical evaluations of maximum comfort postures. The fact that a variety of cost functions has already been used to explain principles of human arm motor control indicates that the CNS does not obey any one particular cost function, but also does not violate general physical and technical principles of optimality, from which particular cost functions come about (Latash, 1993). Hence, additional efforts in searching for new appropriate and effective cost functions are justified. They contribute to a better understanding of biological principles of motor control.

When we speak about *comfort* and *discomfort* we certainly have in mind states closely related to some physiological processes, in particular to *fatigue*, but we are still missing the way to mathematically describe these relations.

### Comfort and Fatigue

The underlying idea of the paper has both theoretical and experimental foundations. Practical experience shows that the human arm commonly takes those postures and executes those movements that are the most comfortable. The term "comfortable" relates to joint positions and engagement of motor units and may also be described by the term "pleasant" (a more precise definition will be given later). On the other hand, endurance contractions of motor units cause muscle fatigue, thus introducing an unpleasant feeling, that is, a sense of discomfort. In everyday life it is easy to observe that after a sensation of discomfort caused by muscle fatigue, the human arm normally reduces engagement of the fatigued motor units, by taking postures that require lower participation of these units. This means that while performing repetitive movements requiring continual repetition of motions in operational space (like in screw-driving tasks), the human arm occasionally reconfigures itself by taking a more comfortable posture, rather than proceeding with some particular pose. The ability

to rearrange its motion is enabled by the presence of both actuator and kinematic redundancy in the human arm (Fuentes & Nelson, 1994). Actuator redundancy comes from the possibility to use several motor units for the same motion of any arm joint. Kinematic redundancy results from the existence of seven degrees of freedom (DOFs) in the arm (from shoulder to wrist), which is more than six independent movements required for an arbitrary positioning and orientation of an object in operational space (Potkonjak, et al., 1998; Sciacivco & Siciliano, 1996). Actuator redundancy and its implementation in robotics are challenging problems that deserve attention. However, they are not considered in this paper, although their role in performing movements in biological mechanisms must be pointed out. Instead, this paper focuses on kinematic redundancy and investigates possibilities to distribute the engagement of robot joints in a human-like fashion, imitating the arm's inherent property to execute comfortable motions. The main objective is to achieve a human-like motion. This can be done if an adequate mechanism is established that simulates biological processes of comfort and discomfort in the arm. It would be useful to rely on relevant findings from already published results of theoretical and experimental investigations. A result which is strongly correlated with our work deals with psychophysical cost functions of joint comfort/discomfort and was presented by Cruse et al. (1990). Their validity was practically justified for arm reach posture prediction by Jung et al. (1994). A psychophysical cost function describes an immediate deviation of joint position from the location of maximum comfort. According to experimental findings given by Cruse et al. (1990), the CNS controls arm motion by minimizing the efforts (from a psychophysical point of view) invested during the movements. Physiological and psychophysical investigations indicated that, in the absence of muscle fatigue, a more comfortable joint pose is closer to the middle of the physiological motion range in that joint. Locally minimizing the function describing a deviation from the position of maximum joint comfort, it is possible to determine comfortable motions of a kinematically redundant mechanism.

Mathematical functions representing current distances from middle positions of joints were used in robotics for joint limits avoidance (Liégeois, 1977; Chan & Dubey, 1995). The applied IK method took care of these distances and forced joint motions to the direction opposite to mechanical boundaries. In this paper, such functions are chosen as starting points in the formulation of an analytic procedure for generating joint movements that are equivalent to the movements of a human arm after appearance of discomfort due to muscle fatigue.

Fatigue in humans is a rather complex issue. On one hand, it is a physiological process related to accumulation of metabolic products. This is the aspect that could be, more or less accurately, modeled. On the other hand, fatigue has a psychological aspect – e.g. a fatigued human feels much better if he simply changes the work he is doing – this aspect cannot be modeled and will not be considered in this paper. Physiological sources of fatigue, although extensively studied, are still not thoroughly known. Basically, fatigue appears after long-standing and powerful contractions of muscle motor units. Increase of lactic acid concentration accompanies the progress of fatigue sensation (pH value decrease in muscle

tissue). Simultaneously, the oxygen distribution is reduced, while concentration of some substances particularly influencing the mechanism of muscle contractions and dilatations decreases, e.g., Adenosine Triphosphate (ATP). As a result, muscular activity declines. The progress in a human's feeling of discomfort due to fatigue grows simultaneously with the progress of fatigue itself. The beginning manifestation is like a slight sense of discomfort in a certain part of the arm, then the discomfort transforms into an unpleasant squib which, finally, results in obtuse pain (Öberg, 1994). Additional engagement of other motor units is then required to sustain the necessary arm actuation.

Muscle fatigue can be quantified by means of objective and subjective methods. Objective methods include mechanical, electromagnetic (EMG), metabolic and physiological measurements (Mizrahi, 1997). Another group of methods is based on the subjective evaluation of the sensed fatigue level, given by the subjects participating in experiments (Öberg, 1994). Because there is a variety of factors indicating the current level of fatigue, it is not possible to distinguish an ultimate method for fatigue quantification. The same statement holds for the models of fatigue, available in literature (see, for example, the work by Kiryu (1998)). However, no matter which method is applied for fatigue quantification, it seems reasonable to consider fatigue as an increasing function. That function is often assumed to be exponential (Peckham, 1972; Vodovnik & Rebersek, 1975; Giat, et al., 1996). The slope of the function depends on the actual engagement of motor units and the current level of fatigue. After some time, saturation appears, as a result of reduced activity of exhausted motor units. An example of a diagram with such characteristics is available in (Jenkins & Quigley, 1992) and corresponds to the increase of lactic acid concentration in a muscle engaged in demanding movements.

Keeping in mind the described principal characteristic of the fatigue function, we will suggest for a mathematical non-dimensional variable to be a measure of fatigue in humans. Later, we will explore eventual meaning, sense, and applicability of such variable in robots, introducing "robot fatigue". The temporal characteristic of robot fatigue must be equivalent to the functional characteristic of biological fatigue, thus opening the possibility of generating human-like motions of the robot joints. The aim is to force a redundant anthropomorphic robot arm to track a given end-effector operational space trajectory, along with producing the most comfortable configurations in the sense of the above mentioned psychophysical cost function. Functionally, robot fatigue will have a response equivalent to the biological muscle fatigue, that is, similar dynamic behavior. Results presented in the rest of the paper will justify this approach. An anthropomorphic seven-DOF human/robot arm performing the screw-driving task will be simulated. It will be shown that the robot arm attains postures and executes motions similarly to that of the human arm performing the same task.

## Mathematical Formulation

### Solution of IK Problem

The arm kinematics will be defined in terms of velocities (Nakamura, 1991; Sciavicco & Siciliano, 1996). The relation

between vectors of configuration (joint) velocities  $\dot{\mathbf{q}}$  and operational (end-effector) velocities  $\dot{\mathbf{x}}$ , is given by the Jacobian form

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}. \quad (1)$$

We assume that redundancy exists, i.e., the number of operational velocities, denoted by  $m$ , is strictly less than the number of configuration velocities, denoted by  $n$ . Normally, it is  $m = 6$  (three translations plus three rotations). For a human arm it holds that  $n = 7$  (3 degrees of freedom (DOF) in shoulder, 2 in elbow, and 2 in wrist). Dimension of the non-square Jacobian matrix  $\mathbf{J}(\mathbf{q})$  is then  $m \times n$ . The redundancy implies a non-unique IK solution, since a given task, defined in terms of operational velocities, can be accomplished with an infinite number of combinations of configuration velocities. We are interested in those joint velocities that would be executed by a human arm in a given task. Adequate velocities can be found by local minimization of the cost function, formed by two quadratic terms (Sciavicco & Siciliano, 1996):

$$\Omega(\dot{\mathbf{q}}) = 0.5 \dot{\mathbf{q}}^T \mathbf{W}' \dot{\mathbf{q}} + 0.5 (\dot{\mathbf{q}} - \dot{\mathbf{q}}_\alpha)^T \mathbf{W}'' (\dot{\mathbf{q}} - \dot{\mathbf{q}}_\alpha). \quad (2)$$

$\mathbf{W}'$  and  $\mathbf{W}''$  denote  $n \times n$  positive definite symmetric weighting matrices, while  $\dot{\mathbf{q}}_\alpha$  represents an  $n$ -component column vector. The first term in eq. (2) enables us to penalize motion of some joints relative to others. In this paper it should provide a distribution of operational motion to the redundant number of arm joints in accordance with the biologically-inspired concept of distributed positioning (DP) (Potkonjak, 1990; Potkonjak & Krstulovic, 1992<sup>a, b</sup>; Potkonjak, et al., 1998), which means stimulating motions of the joints with low inertia and penalizing motions of the joints with high inertia. It should also enable a proper reconfiguration of the arm, in accordance with the progress of fatigue. The second term in eq. (2) aims at the utilization of kinematic redundancy in the sense of a secondary criterion. Minimization of objective (2) subject to constraint (1) is performed by using the method of Lagrange multipliers (Gottfried & Weisman, 1973). Optimal joint velocities are obtained:

$$\dot{\mathbf{q}} = \mathbf{J}_w^\# \dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}_w^\# \mathbf{J}) \mathbf{W}^{-1} \mathbf{W}' \dot{\mathbf{q}}_\alpha, \quad (3)$$

where  $\mathbf{W} = \mathbf{W}' + \mathbf{W}''$  and  $\mathbf{J}_w^\#$  denotes the weighted pseudo-inverse of the Jacobian:

$$\mathbf{J}_w^\# = \mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T)^{-1} \quad (4)$$

Vector  $\dot{\mathbf{q}}_\alpha$  enables the local optimization of some secondary objective function  $G(\mathbf{q})$ , used for the proper utilization of kinematic redundancy. Following (Liégeois, 1977),  $\dot{\mathbf{q}}_\alpha$  is defined as the gradient of  $G(\mathbf{q})$ :

$$\dot{\mathbf{q}}_\alpha = -k_a \left( \frac{\partial G(\mathbf{q})}{\partial \mathbf{q}} \right)^T \quad (5)$$

where  $k_a$  is a scalar coefficient. The final form of the IK solution is obtained by substituting (5) into (3):

$$\dot{\mathbf{q}} = \mathbf{J}_w^\# \dot{\mathbf{x}} - k_a (\mathbf{I} - \mathbf{J}_w^\# \mathbf{J}) \mathbf{W}^{-1} \mathbf{W}' \left( \frac{\partial G(\mathbf{q})}{\partial \mathbf{q}} \right)^T \quad (6)$$



### Choise of the Secondary Objective Function

Definition of the secondary objective function  $G(\mathbf{q})$  providing comfortable motion is discussed in this subsection. The distances of current joint positions  $q_i$ ,  $i=1, \dots, n$ , from the mechanical joint limits  $q_{i,\min}$  and  $q_{i,\max}$  will be the basis for definition of the secondary objective function. In the previous section it was already pointed out that the middle values of human arm joint ranges coincide with positions of the maximum comfort. This fact justifies the choice of  $G(\mathbf{q})$  as a function penalizing deviation from the middle values (Liégeois, 1977; Chan & Dubey, 1995):

$$G(\mathbf{q}) = \frac{1}{2n} \sum_{i=1}^n \left( \frac{q_i - \bar{q}_i}{q_{i,\max} - q_{i,\min}} \right)^2, \quad \bar{q}_i = \frac{q_{i,\max} + q_{i,\min}}{2} \quad (7)$$

or

$$G(\mathbf{q}) = \sum_{i=1}^n \frac{1}{4} \frac{(q_{i,\max} - q_{i,\min})^2}{(q_{i,\max} - q_i)(q_i - q_{i,\min})} \quad (8)$$

### Model of Arm Dynamics

The dynamics of the arm plays an important role in establishing a procedure that provides human-like motions of joints. We adopt the standard representation of arm dynamics:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} + \mathbf{J}^T \mathbf{F} \quad (9)$$

where  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  denote  $n \times 1$  vectors of joint positions, velocities, and accelerations;  $\mathbf{H}$  is an  $n \times n$  inertia matrix;  $\mathbf{h}$  is an  $n \times 1$  vector of centripetal, Coriolis', friction and gravitational torques;  $\boldsymbol{\tau}$  denotes the  $n \times 1$  vector of driving torques in the joints;  $\mathbf{J}$  is the Jacobian, and  $\mathbf{F}$  is the  $m$ -component external force/torque. Driving torques are produced by actuators – muscles in humans and motors in robots.

### Model of Fatigue Process

This subsection suggests a method that should be used to simulate the effects of physiological fatigue in human muscles.

The key features of the fatigue function were revealed in subsection “*Comfort and Fatigue*”. We now look for a proper mathematical model to express these features. We first introduce a non-dimensional variable being the measure of fatigue. It will be called simply *fatigue*. Let  $z_i$  be the fatigue in joint  $i$  ( $i=1, \dots, n$ ). We assume that the level of fatigue directly depends on the accumulation of metabolic products – concentration of lactic acid. So, the concentration could be considered as fatigue, after normalizing it to become a non-dimensional variable.

We consider fatigue as an accumulation process. Like in modeling any accumulation process, we will consider the gradient of the state coordinate as balancing input and output. A good paragon is the process of heating a body, being the accumulation of thermal energy: temperature  $\theta$  is the state variable,  $R\dot{\theta}$  is the input from electric heater,  $k(\theta - \theta_0)$  is the output (energy transition body-to-ambient), and the model is finally  $C\dot{\theta} = R\dot{\theta} - k(\theta - \theta_0)$ , where  $C$  is the specific thermal capacity. With the fatigue process,  $z_i$  is the state variable. Let us discuss the input and the output.

- One notes that there exists an input, a source of lactic acid – these are metabolic processes which intensify with the stronger muscle activity i.e. with the higher joint torque  $\tau_i$ . There are no reliable results revealing how the production of acid depends on the torque  $\tau_i$ . In order to develop the methodology, we follow the paragon and adopt a quadratic function. So, the input rate can be expressed as  $A_i \tau_i^2$ , where  $A_i$  is the coefficient that should be determined experimentally.

- Next, one notes that there is also an output – blood takes the acid out of the muscle. The output rate can be considered proportional to the difference in concentration between the muscle ( $z_i$ ) and the blood ( $z_{i,0}$ ), and so it is:  $K_i(z_i - z_{i,0})$ , where  $K_i$  is the conductance (transition coefficient).

Now the balance is:

$$M_i \dot{z}_i = A_i \tau_i^2 - K_i(z_i - z_{i,0}) \quad (10)$$

where  $M_i$  is the specific accumulation capacity. The model can be rewritten to the form

$$T_i \dot{z}_i = \frac{A_i}{K_i} \tau_i^2 - (z_i - z_{i,0}) \quad (11)$$

revealing the time constant  $T_i$ .

Model (11) considers  $z_{i,0}$  as being constant. This means that we neglect the accumulation process in the blood. If one wishes to expend the fatigue dynamics to include bloodstream system, he should note that the acid taken from the muscle should be treated as input to the blood. The output should be defined as the place where the “cargo” is unloaded (liver). The concentration  $z_{i,0}$  now becomes a new state variable and the fatigue process becomes of the second order.

### The Full Model

The full model means the set of equations that can be numerically integrated to calculate the system behavior in all its aspects. So, we talk about simulation and it goes in few steps:

- We start from the fact that a given task means a prescribed motion in operational space,  $\mathbf{x}(t)$ .
- IK is resolved by applying expression (6) along with (8). This way, the joint-space motion  $\mathbf{q}(t)$  is obtained which satisfies the request for comfort.
- With known joint motions, the joint torques  $\boldsymbol{\tau}(t)$  are found from the dynamic model (9).
- Finally, the progress of fatigue  $\mathbf{z}(t)$  is calculated by integrating the model (11).

What still remains an open question is: does and how fatigue influences the above formulated simulation procedure.

### How Fatigue Influences the Arm Motion

Here, we suggest the proper means for establishing the functional dependence between fatigue time history  $z_i(t)$ ,  $i = 1, \dots, n$ , and robot arm motions. For this purpose, we should reinstate the expected effects of fatigue progress. During manipulation, a human arm performs movements adequate to the desired manipulation task, permanently accommodating its configurations to the actual level of muscle fatigue. Present kinematic and actuator redundancy allows execution of the



manipulation task in a comfortable way, by appropriate distribution of joint motions and participation of different motor units. In such ways exhausted muscles may recover and other muscles increase their activity. A similar strategy could be applied to a robotic arm. Note that capabilities of the kinematic redundancy will be utilized only – actuator redundancy is out of scope of this paper.

### Reconfiguration

For each arm joint it is necessary to specify an appropriate critical level of the fatigue:  $z_{i,cr}$ . It is the level when a human starts to feel unpleasant sensations in the considered joint. Thus,  $z_{i,cr}$  should not be seen as a definite limit but rather a bound of a desired region of working mode. If the fatigue  $z_i$  is less than  $z_{i,cr}$ , a human feels fine and will continue working in the same way. If  $z_i$  exceeds the critical value, the human can still work in the same way but he will not feel comfortable. To prevent the discomfort, a proper depression of motion in the critical joint is needed. It is necessary to reconfigure the arm by taking a posture that will engage other joints more thus giving the exhausted joint a chance to rest. Note that reconfiguration is done in the joint space and does not affect the operational motion  $\mathbf{x}(t)$  and the execution of a given task. Reconfiguration of joint motions can be achieved on the IK level, by means of the proper weighting matrices. We remind that there are two matrices making a sum  $\mathbf{W} = \mathbf{W}' + \mathbf{W}''$ . Matrix  $\mathbf{W}'$  should provide human-like distribution of joints motions (DP concept) and an adequate reconfiguration of the arm with respect to actual levels of joint fatigue. By means of  $\mathbf{W}''$  one can specify higher engagement of some joints in realization of the secondary objective. In this paper, it is assumed that all joints have equal priority in realization of the secondary objective. In such a way, the role of  $\mathbf{W}'$  determines a particular choice of  $\mathbf{W}$ . To ensure the proper reconfiguration, penalty functions are introduced into the weighing matrix:

$$\mathbf{W} = \text{diag}[\varphi_1(z_1), \dots, \varphi_n(z_n)] \quad (12)$$

Penalty functions  $\varphi_i(z_i)$  should penalize the exhausted joints and stimulates those that are still “fresh”. Mathematically speaking,  $\varphi_i(z_i)$  should be constant until  $z_i$  reaches  $z_{i,cr}$  and monotonically increasing above  $z_{i,cr}$ . In this way, the penalty functions will contribute to reduced movement of each joint in which the actual value of fatigue exceeds an assigned critical level. The choice of a particular penalty function is task dependent. For the simulation study of this article we adopt a quadratic function:

$$\varphi_i(z_i) = \begin{cases} w_i, & z_i < z_{i,cr} \\ w_i + k_i(z_i - z_{i,cr})^2, & z_i \geq z_{i,cr} \end{cases}, \quad (13)$$

where the initial weighing factor  $w_i$  is a scalar constant and the coefficient  $k_i > 0$  determines the desired slope of the penalty function.

It is expected that reduced engagement of the exhausted joints will give them a chance to rest and go out of the critical working mode. Several reconfigurations may happen, one after the other, as different joints reach the critical levels. If the imposed task is not too tough, the arm will finally finds a

steady state in which it can operate for a longer time. We remind that reconfigurations do not affect the execution of the task since it does not reflect in the operational space  $\mathbf{x}$ .

### Degeneration

If the task is too demanding, it may happen that, in spite of reconfiguration, the fatigue functions  $z_i(t)$  continue to rise. This means that reconfiguration delays the fatigue problem but does not eliminate it. To handle this situation, some upper limits of fatigue are adopted:  $z_{i,max}$ ,  $i = 1, \dots, n$ . The limit in the  $i$ -th joint,  $z_{i,max}$ , determines the level when discomfort in the joint turns into pain that cannot be endured. In this situation, a further rise of fatigue must be prevented regardless of the consequences, even if the execution of the task is compromised. Hence, we call this phase degeneration. The over-exhausted joint must rest and we emulate this process by using a “torque limiter”. The limiter will allow the torque that is smaller than the required value by the factor  $D$ , and thus for the joint  $i$  it will be:

$$\tau_i = D_i(z_i)\tau_i^{req} \quad (14)$$

where  $\tau_i$  is the actual torque and  $\tau_i^{req}$  is the value required by the dynamics of the given task. The damping factor  $D_i(z_i)$  depends on the actual level of fatigue. In order to efficiently relax the over-exhausted joint, an exponentially decreasing function is adopted:

$$D_i(z_i) = \begin{cases} 1, & z_i \leq z_{i,max} \\ e^{-(z_i - z_{i,max})}, & z_i > z_{i,max} \end{cases} \quad (15)$$

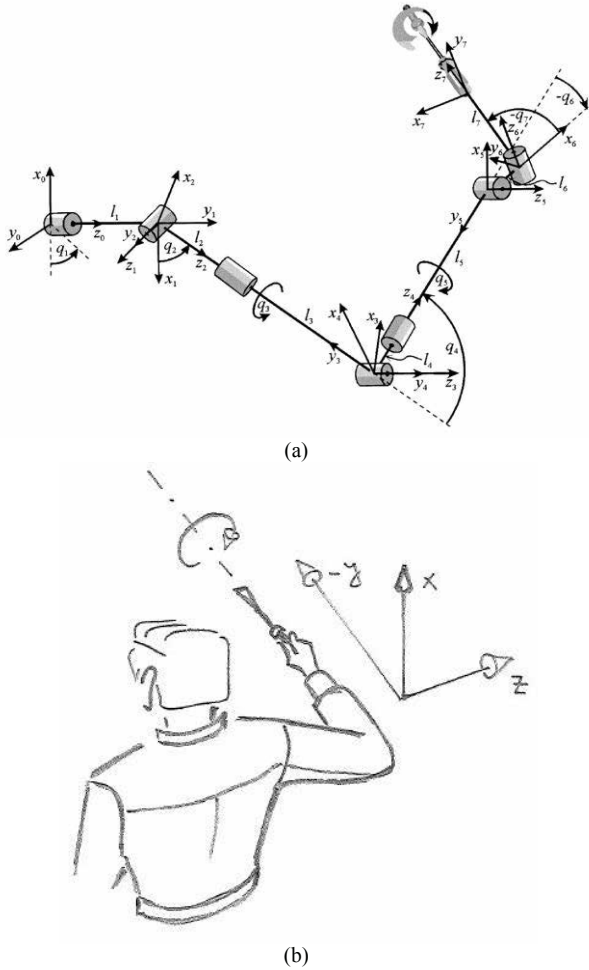
Damping the torque will result in insufficient joint drive and accordingly in the degeneration of motion trajectories, in both joint ( $\mathbf{q}$ ) and operational ( $\mathbf{x}$ ) space. Thus, the task is no more executed properly.

### Implications to a Robotic Arm

The above discussion was mainly focused on human arm but with the idea in the background to find a proper interpretation and implication in robots. The answer is generally in the possibility to achieve a human-like behavior of robot. Potkonjak et al. (2002<sup>a, b</sup>, 2005) suggested approaching this problem from the aspect of human-robot communication, and contributing to gestural communication. The research paid particular attention to generation of a nonverbal message about overloading. The thermal dynamics, that is, robot motors heating, was considered and the rise of temperatures was used as the measure of “robot fatigue”. Redistribution of joint engagements was suggested as the solution which would relax the overheated motors. This reconfiguration would be observable to people being around and thus would be a nonverbal message about fatigue and exhaustion. The authors mainly concentrated on robots engaged in fine-motor-control tasks and particularly handwriting. The present research shares some basic ideas but it generalizes the problem by putting it in a wider context of mathematical modeling of physiological processes and human-robot analogy.

## Example – Screw-Driving Task

Effects of the suggested method are analyzed for a task very demanding regarding applied force and torque – the screw-driving task. A human and/or an anthropomorphic robot arm has to screw the bolt into the hole on the vertical work surface (wall). Although this operation often involves a specific motorized tool, an electric screw-driver, we here consider screw-driving as a purely manual operation. This means that, apart from enabling the proper position of the “old-fashion” screwdriver in the space, the arm should also provide angular screw-driving movements about the longitudinal screwdriver’s axis (Fig. 1).



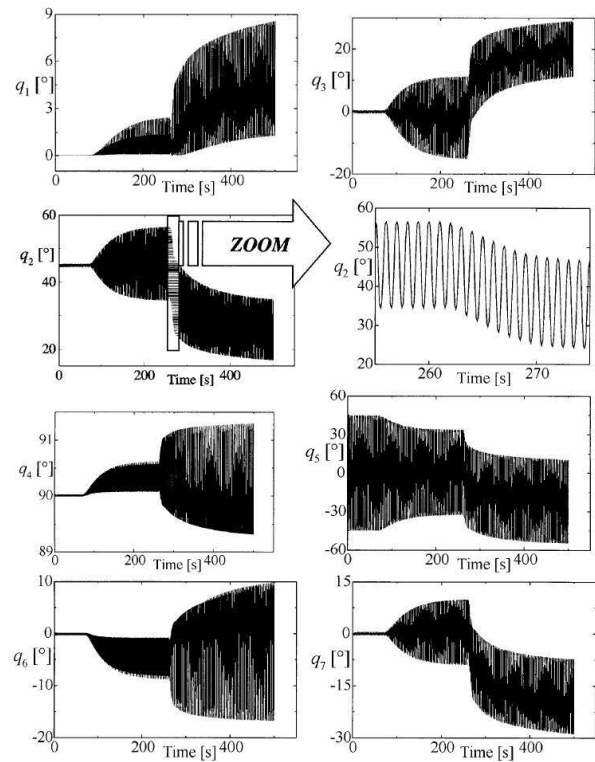
**Fig. 1.** Redundant arm in screw-driving task: (a) configuration, (b) initial position

The considered arm configuration is shown in Fig. 1a. It has seven DOFs, meaning that it is redundant for the given task that requires six. Initial position is shown in Fig. 1b.

Initial position of the robot arm is defined by:  $q(t=0) = (0, 45^\circ, 0, 90^\circ, -45^\circ, 0, 0)$ . Accordingly, the initial arm posture has a stretched wrist and the forearm and the screwdriver are aligned perpendicularly to the wall.

A screw-driving task requires rotational motion of a screw about its longitudinal axes, along with keeping its

perpendicularity to the wall. It is assumed that screwing consists of a series of single revolute movements (forward and backward), each of  $\pi/2$  [rad]. A forward rotation (screwing in) is indicated in Fig. 1. Backward rotation does not drive the screw, but only prepares the screwdriver for the next turn. Each movement takes  $T=0.5$ s. Different studies investigating motor control of human movements have shown that human arm performs smooth voluntary movements with bell-shaped velocity profiles (for illustration, see (Hogan, 1984)). In our simulations the bell-shaped velocity profile was approximated with a cosine velocity profile (Vukobratovic & Kircanski, 1986) and applied to rotational motions of the screwdriver. This way we complete the definition of the end-effector motion task. The full task, however, includes the force and the torque which the screwdriver applies to the screw – the arm motor units have to provide longitudinal force  $F=50$ N, and the torque  $M=6$ Nm about the longitudinal axis. The force and the torque are applied only while turning the screwdriver forward – the backward rotation is relaxed.



**Fig. 2.** Joint motions:  $q_i(t)$ ,  $t = 1, \dots, 7$ . Zoomed view of  $q_2$  shows the motion drift in shoulder: upper arm starts to move down toward the trunk.

We now apply the derived models of fatigue progress and the suggested method for IK resolution, with the aim to simulate the arm behavior in the imposed task. The calculated behavior will then be compared with the behavior observed with a control group of 5 human study subjects. Note that the model parameters used in simulation were tuned (using simulation experiments) so as to stimulate the system to feature relevant effects earlier, thus eliminating the need for too long simulation. Hence, the parameters used in simulation

differ from those of study subjects. Monitoring of the control group was not based on some measurement but rather on the visual observation and the verbal descriptions given by the study subjects. Therefore, the simulation results and the experiment can be compared on the qualitative level only.

The joint motions,  $q_i(t), t = 1, \dots, 7$ , obtained by simulation are shown in Fig. 2 (note that the applied order 1, 3, 2, ... instead of regular 1, 2, 3, ... uses the space in the figure more economically). Note that the total time of monitoring the event was 500 s. Since a cycle of screwing-in (forward-plus-backward rotation) lasts 1 s, it is not possible to observe a particular oscillation in joint motions. One can only see the envelope, but it is sufficient for understanding the results. The only exception is the zoomed diagram for joint 2, where oscillations are visible. Bearing in mind the arm starting position, it is clear why the screw driving is initially performed by joint 5 alone. It is the only joint able to provide the rotational motions about the longitudinal axis of the screwdriver.

The simulated progress of fatigue,  $z_i(t), i = 1, \dots, 7$ , is presented in Fig. 3.

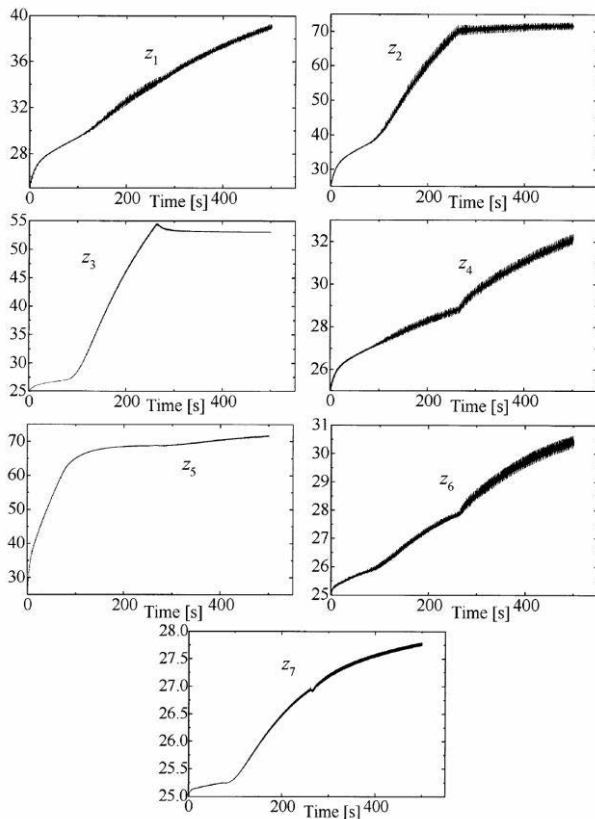


Fig. 3. Progress of fatigue,  $z_i(t), t = 1, \dots, 7$ .

Relations between time histories of  $q_i(t)$  and  $z_i(t)$  should be considered next. This will be done by comparing diagrams from Figs. 2 and 3. Let us start from joint 5. Diagram  $q_5$  in Fig. 2 shows large turns which generate the screwing torque. Hence, fatigue  $z_5$  (Fig. 3) progresses fast. When  $z_5(t)$  reaches the assigned critical level  $z_{5,cr} = 60$ , penalty function  $\varphi_5(z_5)$

starts to work, causing reduced engagement reflecting in decreased amplitude of motion in diagram  $q_5(t)$  (Fig. 2).  $z_5$  features a reduction of slope. The change in slope is not sharp, but comes some time after reaching  $z_{5,cr}$ , resembling human reaction. So, with appearance of fatigue symptoms in the joint 5, its engagement decreases. This is enabled by proper participation of other joints, which starts exactly at the moment when joint 5 reaches its critical value. This effect is apparent from all diagrams  $q_i(t)$ .

Diagrams of motions in joints 2, 3, and 7 deserve special attention, because of their prevalent participation in compensating the reduced involvement of joint 5. This is also similar to the natural behavior of human arm, which after sensing the discomfort (fatigue) in the forearm, engages exactly the same joints to relax exhausted muscles. Increased engagement of these joints results in increase of their fatigue. Two of the three most actively participating joints (2 and 3) are in the shoulder. These joints, besides other roles, compensate the gravity load of the complete arm. The third active rotation (joint 7) is in the wrist and it coincides with that rotation of a human wrist which is able to participate in endurance movements. Joint 7 can stand significant dynamical demands and slowly fatigues. Its engagement in providing speeded-up motions in the writing task has been investigated in (Potkonjak, et.al, 1998). These facts imply that fatigue effects should appear in the shoulder first, rather than in the wrist. This is equivalent to the natural behavior of a human arm. As a consequence, fatigue  $z_2$  is the next (after  $z_5$ ) to reach its adopted threshold  $z_{2,cr} = 70$ . After reaching the threshold, the penalty function will keep the fatigue in the vicinity of that level. The motion in joint 2 (shoulder) drifts toward lower values of  $q_2$ . This is shown in Fig. 2 and zoomed for better observation. The suggested method for redundancy resolution based on actual fatigue provides reconfiguration of the arm mechanism in a way identical to a human arm after appearance of biological fatigue. The arm puts its elbow closer to the trunk, after subject to fatigue caused by endurance movements during screw driving. This new posture of a human arm is more comfortable to work. It is important to note that the arm proceeds its normal operation (in the sense of task execution), just taking the new posture. Redistribution of motion, depression of some joint motions and stronger engagement of others, does not compromise the end-effector motion. The complete redistribution is shown in diagrams  $q_i(t), t = 1, \dots, 7$  (Fig. 2).

In the above discussion, the statements that some behavior calculated by simulation resembled the human behavior, were based on the qualitative comparison between the simulation results and the observation from the human control group. The most visible reaction of human study subjects was turning the elbow down to the trunk. This is also visible in the simulation diagram of  $q_2$  in Fig. 2.

## Conclusion

The objective of the paper was to explore how physiological processes, in particular fatigue, influence human motion, with the idea of formulating mathematical models describing this relation. Resolution of the inverse kinematics of redundant



arm was discussed first, proposing a biologically inspired method that took care of the comfort of motion and utilized the actual level of fatigue in arm motor units. The method allowed the reconfiguration of the arm that gave the fatigued joints a chance to rest by engaging more the joints which were “fresher”. In order to simulate the progress of fatigue, mathematical model of fatigue was derived based on a general model of accumulation processes. The developed methods were tested by simulation and qualitative comparison with the observed behavior of the human control group. Results obtained by simulation featured a human-like behavior that qualitatively agreed with the observation from study subjects. Implications of the results to anthropomorphic robots were indicated.

## References

- Holland, O., Knight, R. (2006). The Anthropomorphic Principle. *Proceedings of the AISB'06 Symposium on Biologically Inspired Robotics*, pages 115–122. Bristol, UK.
- Diamond, A., Knight, R., Devereux, D., Holland, O. (2012). Anthropomorphic Robots: Concept, Construction and Modelling. *International Journal of Advanced Robotic Systems*, 9:1–14.
- Potkonjak, V., Svetozarevic, B., Jovanovic, K., Holland, O. (2011). The Puller-Follower Control of Compliant and Noncompliant Antagonistic Tendon Drives in Robotic Systems. *International Journal of Advanced Robotic Systems*, 8(5):143–155.
- Wittmeier, S., Alessandro, C., Bascarevic, N., Dalamagkidis, K., Diamond, A., Jantsch, M., Jovanovic, K., Knight, R., Marques, H.G., Milosavljevic, P., Svetozarevic, B., Potkonjak, V., Pfeifer, R., Knoll, A., Holland, O. (2013). Toward Anthropomorphic Robotics: Development, Simulation, and Control of a Musculoskeletal Torso. *Artificial Life (MIT press)*, 19(1):171–193.
- Mizuuchi, I., Nakanishi, Y., Sodeyama, Y., Namiki, Y., Nishino, T., Muramatsu, N., Urata, J., Hongo, K., Yoshikai, T., Inaba, M. (2007). An advanced musculoskeletal humanoid Kojiro. *Proceedings of the 7th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2007)*, pages 294–299. Pittsburgh, USA.
- Sodeyama, Y., Nishino, T., Namiki, Y., Nakanishi, Y., Mizuuchi, I., Inaba, M. (2008). The Designs and Motions of a Shoulder Structure with a Spherical Thorax, Scapulas and Collarbones for Humanoid “Kojiro”. *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008)*, pages 1465–1470.
- Haggard, P., Libet, B. (2001). Conscious Intention and Brain Activity. *Journal of Consciousness Studies*, 8(11):47–63.
- Pfeifer, R., Bongard, J. (2007). *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press, Cambridge, MA.
- Potkonjak, V., Tzafestas, S., Kostic, D., Djordjevic, G., Rasic, M. (2003). Illustrating man-machine motion analogy in robotics - The handwriting problem. *IEEE Robotics and Automation Magazine*, 10(1):35–46.
- Hollerbach, J. M., Suh, K. C. (1987). Redundancy resolution of manipulators through torque optimization. *IEEE Journal of Robotics and Automation*, 3(4):308–316.
- Khatib, O. (1983). Dynamic control of manipulators in operational space. *Proceedings of the Sixth World Congress on Theory of Machines and Mechanisms*, pages 1123–1131. New Delhi, India.
- Hogan, N. (1984). An organizing principle for a class of voluntary movements. *The Journal of Neuroscience*, 4(11):2745–2754.
- Seif-Naraghi, A. H., Winters, J. M. (1989). Changes in musculoskeletal control strategies with loading: Inertial, Isotonic, Random. *Proceedings of the ASME Biomechanics Symposium*, pages 355–358. San Diego, USA.
- Cruse, H., Wischmeyer, E., Bruwer, M., Brockfeld, P., Dress, A. (1990). On the cost functions for the control of the human arm movement. *Biological Cybernetics*, 62(6):519–528.
- Latash, M.L. (1993). *Control of human movement*. Human Kinetics Publishers, Champaign, Illinois.
- Fuentes, O., Nelson, R.C. (1994). Morphing Hands and Virtual Tools (Or What Good is an Extra Degree of Freedom?), Technical Report, 551, Department of Computer Science, The University of Rochester.
- Potkonjak, V., Popovic, M., Lazarevic, M., Sinanovic, J. (1998). Redundancy problem in writing: From human to anthropomorphic robot arm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 28(6):790–805.
- Sciavicco, L., Siciliano, B. (1996). *Modeling and Control of Robot Manipulators*. McGraw-Hill, London, UK.
- Jung, E.S., Choe, J., Kim, S.H. (1994). Psychophysical cost function of joint movement for arm reach posture prediction. *Proceedings of the Human Factors and Ergonomics Society 38th Annual Meeting*, pages 636–640. Santa Monica, CA.
- Liégeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(12):868–871.
- Chan, T.F., Dubey, R.V. (1995). weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Transactions on Robotics and Automation*, 11(2):286–292.
- Öberg, T. (1994). Subjective and objective evaluation of shoulder muscle fatigue. *Ergonomics*, 37(8):1323–1333.
- Mizrahi, J. (1997). Fatigue in muscles activated by functional electrical stimulation. *Critical Reviews in Physical and Rehabilitation Medicine*, 9(2):93–129.
- Kiryu, T., Morishita, M., Yamada, H., Okada, M. (1998). A muscular fatigue index based on the relationships between superimposed M wave and preceding background activity. *IEEE Transactions on Biomedical Engineering*, 45(10):1194–1204.
- Peckham, P. H. (1972). Electrical Excitation of Skeletal Muscle: Alterations in Force, Fatigue and Metabolic Properties, Technical Report, EDC 4-72-32, Case Western Reserve University.
- Vodovnik, L., Rebersek, S. (1975). *Electrical Stimulation as a Rehabilitation Method to Improve Abnormal Locomotion and Manipulation: Final Report*, University of Ljubljana.
- Giat, Y., Mizrahi, J., Levy, M. (1996). A model of fatigue and recovery in paraplegic’s quadriceps muscle subjected to intermittent FES. *Journal of Biomechanical Engineering*, 118(3):357–366.
- Jenkins, D. G., Quigley, B. M. (1992). Endurance training enhances critical power. *Medicine and Science in Sports and Exercise*, 24(11):1283–1289.
- Nakamura, Y. (1991). *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley, Amsterdam, Netherlands.
- Potkonjak, V. (1990). Distributed positionong for redundant robotic systems. *Robotica*, 8(1):61–76.
- Potkonjak, V., Krstulovic, A. (1992). Mathematical modeling of a redundant anthropomorphic arm (part I). *Robotics and Autonomous Systems*, 9(3):165–170.
- Potkonjak, V., Krstulovic, A. (1992). Simulation of a redundant anthropomorphic arm (part II). *Robotics and Autonomous Systems*, 9(3):171–179.
- Gottfried, B. S., Weisman, J. (1973). *Introduction to Optimization Theory*. Prentice-Hall, Englewood Cliffs, NJ.
- Potkonjak, V., Tzafestas, S., Radojicic, J., Kostic, D. (2002). Modeling Robot “Psycho-Physical” State and Reactions – A New Option in Human-Robot Communication, Part 1: “Concept and Background”. *Journal of Intelligent and Robotic Systems*, 35(4):339–352.
- Potkonjak, V., Tzafestas, S., Radojicic, J., Kostic, D. (2002). Modeling Robot “Psycho-Physical” State and Reactions – A New Option in Human-Robot Communication, Part 2: “Modeling and Simulation”. *Journal of Intelligent and Robotic Systems*, 35(4):353–364.
- Potkonjak, V. (2005). Robotic Handwriting. *International Journal of Humanoid Robotics*, 2(1):105–124.
- Vukobratovic, M., Kircanski, M. (1986). *Kinematics and Trajectory Synthesis of Manipulation Robots*. Springer-Verlag, Berlin, Germany.

# An artificial lizard regrows its tail (and more): regeneration of 3-dimensional structures with hundreds of thousands of artificial cells

Alessandro Fontana<sup>1</sup> and Borys Wróbel<sup>1,2</sup>

<sup>1</sup>Evolving Systems Laboratory, Faculty of Biology, Adam Mickiewicz University, Poznań, Poland, and

<sup>2</sup> Systems Modelling Laboratory, IO PAS, Sopot, Poland  
fontana@evosys.org, wrobel@evosys.org

## Abstract

Biological multicellular structures can not only self-generate from a single cell but also self-regenerate after damage. In this paper we investigate self-regeneration in a model of artificial development, Epigenetic Tracking. 3-dimensional cellular structures grown using our model reach a size and a level of complexity unmatched by other models in the field, thanks to several features of Epigenetic Tracking. One of these features is that only a small fraction of cells in the body, called drivers, orchestrate development. In this paper we use the mechanism for the generation of drivers based on the diffusion of morphogens as a foundation of several new mechanisms in Epigenetic Tracking, and show that these mechanisms allow for self-regeneration after removal of arbitrarily large portions of the multicellular body.

## Introduction

Models of development can be divided into two broad classes: grammatical and chemical. The first class includes L-systems, introduced by Lindenmayer (10) to model plant growth, and models based on context-free or context-sensitive grammars, instruction trees or directed graphs (e.g., 1; 7; 8). Grammatical models can generate surprisingly complex, life-like shapes, even though they do not include mechanisms corresponding to the biological processes working at the molecular level. In contrast, chemical models (e.g., 2; 9; 11) do include such mechanisms, inspired by information processing inside cells (gene regulatory networks) and by communication between cells (diffusion of chemical substances; considered already by Alan Turing, 1952). Although some models of development can be considered either grammatical or chemical, the division between these classes is fuzzy, and much can be achieved by combining features of both to bring computational efficiency on one hand, and biological plausibility on the other. Such combination of features stands behind Epigenetic Tracking (3).

In Epigenetic Tracking, self-generation of 3-dimensional multicellular structures starts from a single cell containing a genome that encodes all the information necessary to direct development. The genome can be evolved using a genetic algorithm with a fitness function measuring the proximity of

the final structure (the body) to a target shape. The complexity of the bodies obtained with Epigenetic Tracking (number of fine morphological features and patterning) and their size (reaching millions of cells) has not been matched yet by any other model. Self-generation of such large and complex bodies is possible thanks to the division of cells into normal cells and drivers: a small fraction of cells that orchestrate development.

We have recently introduced a new mechanism into Epigenetic Tracking, a mechanism for the generation of drivers, based on the diffusion of chemical substances (morphogens; 6). In the present paper we introduce several additional new mechanisms, guided by the assumption that regeneration replays the events that occurred during development (13, see also 2). We show that these additional mechanisms allow to add self-repair to the list of phenomena we previously investigated in our system (ageing and cancer, 4; and the hypothetical transfer of information between somatic cells and the germline, 5; Fontana and Wróbel).

## Epigenetic Tracking: a model of evolving, self-generating multicellular structures

In Epigenetic Tracking multicellular bodies consist of cube-shaped cells on a 3-dimensional grid. The growth starts from a single cell and continues through a pre-specified number of developmental stages. Cells belong to two categories: *normal* and *drivers*. Each driver has an associated array of digits, called *mobile code*. All the cells carry the same genome, an array of characters (from a 4-letter alphabet). The mobile code can be considered as an abstraction for the set of regulatory factors present in a cell: it allows drivers to behave differently despite sharing the same genome.

The genome consists of developmental genes, which all have a left part and a right part. The left part contains three fields: *switch*, which specifies if a gene is active or inactive, *timer*, and *mobile sequence*. At each developmental stage, the mobile sequence of each developmental gene is compared with the mobile code value of each driver, and the timer is compared with the value of the current developmental stage. If both match, the right part of the gene fires. Each



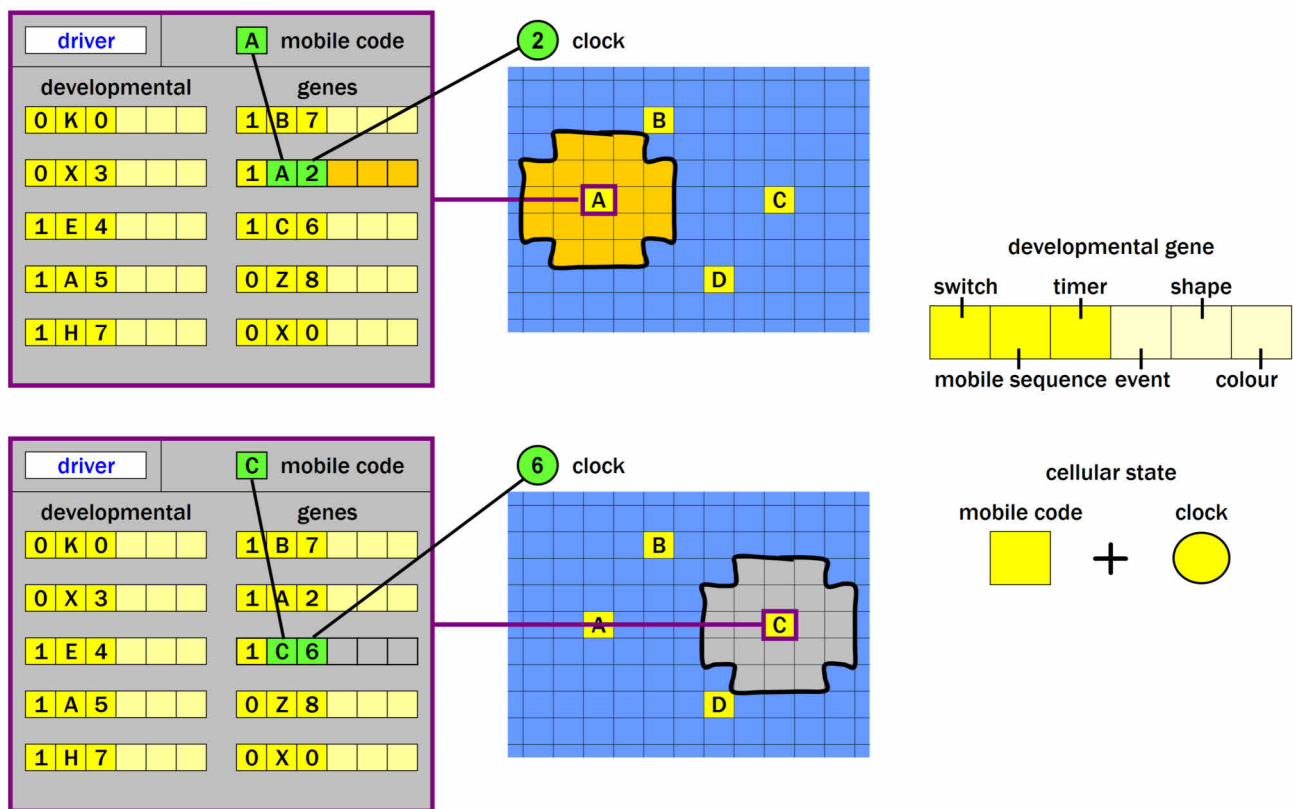


Figure 1: Orchestration of developmental events by drivers in Epigenetic Tracking. On the top, a driver (one of the cells in yellow) with mobile code A triggers a proliferation event at developmental stage 2. On the bottom, a driver with mobile code C triggers a cell death event at stage 6. Black lines indicate the match of the fields in the left part of the genes with the mobile codes of the drivers and the clock. The shape of the created or deleted cell masses is encoded in the right parts of genes. The schematic genome has 10 genes.

right part has three fields. One field determines the type of event – local proliferation or death (removal of cells from the grid; Fig. 1). Another field specifies the shape of the local structure created by proliferation or removed by cell death. The third field determines the phenotype of the normal cells produced in case of proliferation (which is represented by their colour).

A proliferation produces normal cells and drivers. Drivers are placed among normal cells after proliferation, and are much fewer than normal cells in number. Each new driver obtains a new and unique mobile code. The creation of drivers in the experiments described in this paper relies on the diffusion of morphogens, belonging to a finite number of types (we recently implemented this mechanism in Epigenetic Tracking, 6). After a driver orchestrates a developmental event, it persists in the structure and becomes a source of the morphogen which had the lowest concentration at the driver's position when this cell was activated; the diffusion of this morphogen will contribute to the chemical landscape in the body and influence the creation of future

drivers (Fig. 2).

The concentration of a particular morphogen ( $C$ ) in the body follows the equation  $C = 9 - \text{round}(D/G)$ , where  $D$  is the Euclidean distance between a given position on the grid and the closest driver that produces the morphogen, and  $G$  is a system parameter. If the formula gives a negative number, the concentration is set to 0. Because of the rounding, the concentrations can take an integer value between 0 and 9, and positions close-by can have the same concentrations of all morphogens. The cell in the centre of each such region (determined by averaging the coordinates of all the cells there) is a candidate for becoming a new driver. The drivers are created after sorting the regions by size, in the largest region first, provided that each new driver is sufficiently far away from the closest existing driver (this distance is also a system parameter).

The mobile code of the new drivers is derived from the code of the driver that created it through proliferation, but also includes information about the concentration of morphogens in the region. To do so, the mobile code is sep-

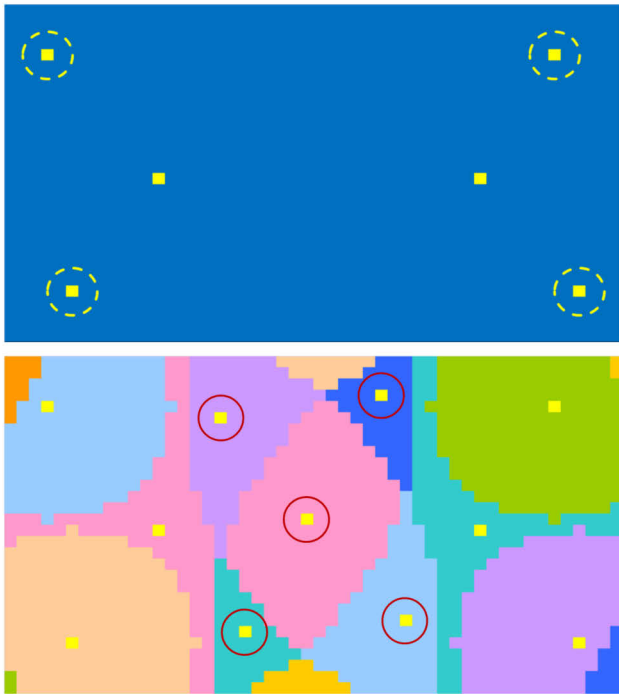


Figure 2: Generation of new drivers based on diffusion of morphogens. Driver cells which have orchestrated an event (yellow cells circled in top panel) persist after doing so in the structure and produce morphogens. There is a finite number of morphogens and their concentrations are rounded, so the structure divides up into multicellular regions (each marked with a different colour in the bottom panel) having the same concentrations of morphogens. New drivers (indicated with red circles) form from the central cells in such regions, if these cells are sufficiently distant from any other driver.

arated into as many sub-fields as there are developmental stages, and the concentrations are encoded in the sub-field corresponding to the current stage. For example, if the concentration values are [4,7,8,2], the number encoded in this sub-field (using a 4-digit positional code) will be 4782. The parameters of the system can be varied to ensure a sufficiently high driver density as to preserve evolvability (6).

Because the first proliferation originates from one initial driver cell (the zygote), morphogen regions are absent, and another mechanism is needed at this point to form new drivers: they are placed using a pre-specified pattern. This initial placement of drivers can be seen as deriving from morphogen gradients in the egg itself (maternal factors, 13).

#### Four new mechanisms in Epigenetic Tracking to allow for self-regeneration

Because of the central assumption of our model of regeneration (that the state of the body at the start of regeneration has

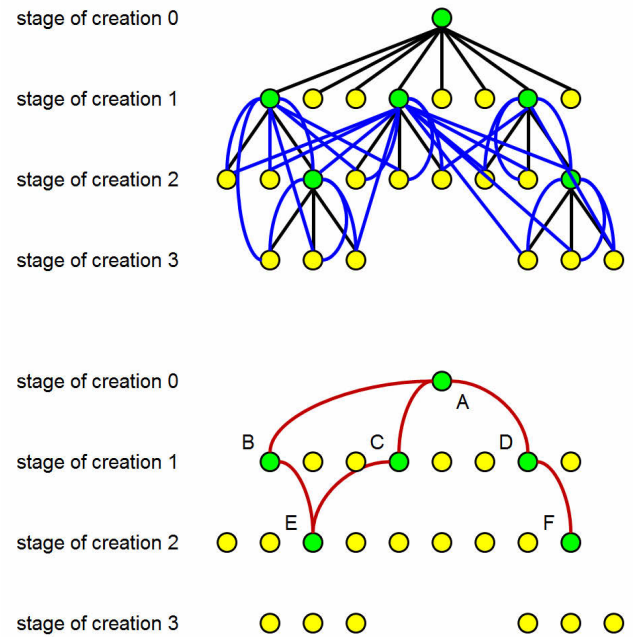


Figure 3: Dependencies among drivers. The cells turned into drivers are originally produced in proliferations orchestrated by other drivers (these dependencies are represented by black lines on the top panel), but their formation is influenced by morphogens produced by other drivers (blue lines), with the exception of drivers created in the first stage (which are not created using morphogens; see text for details). This information can be used to outline a dependency graph (bottom panel) between drivers which were activated during development (shown in green), and can be reactivated for regeneration. For example driver E depends on driver cells B, and A.

to be similar to some state during embryonic development), the drivers that orchestrated development need to persist in the body – they will be needed to orchestrate the events during regeneration. These drivers, once they trigger a developmental event, are kept in a deactivated state (so that this event is not triggered again). This is the first new mechanism introduced in our system.

The second mechanism permits detection of damage. All cells created in a proliferation event send chemical signals to the deactivated driver which created them. If many of these cells are destroyed, this driver can be reactivated because it receives less of these signals. A large damage can result in the reactivation of many drivers.

Before such reactivation can occur, however, the debris left by the damage needs to be removed, so that the structure contains no cells created during development by the driver that is reactivated, or indeed any cells whose creation depended on such a driver (using a dependency graph like

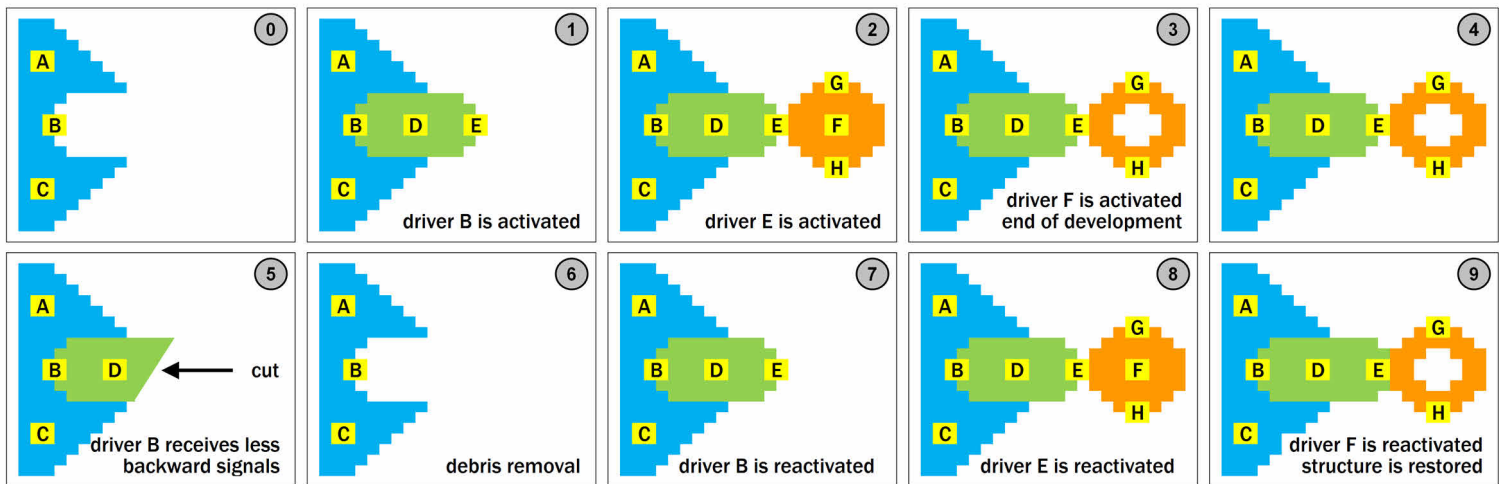


Figure 4: Regeneration of a body part. Panels (and stages) 0-4: the developmental trajectory for a hypothetical body part. After the end of development (stage 4), a portion of the structure is cut at stage 5. After debris removal (stage 6), driver B is reactivated, leading to the recreation of drivers E, which is also reactivated, and so on, until the structure is completely regrown.

the one in Fig. 3). Should such cells remain, some parts of the structure could be duplicated or regeneration would not work correctly. This debris removal is the third new mechanism.

The fourth mechanism recreates exactly the same landscape of morphogens as during development, by excluding from the sources of morphogens all the drivers activated originally (during development) after the driver that is now reactivated during regeneration. For example, if the reactivated driver was originally activated at stage 5, a driver activated at stage 12 would be excluded.

Finally, we needed to deactivate one of the mechanisms that were present in previous versions of Epigenetic Tracking: in the version presented here, proliferation does not cause the cells present at this point in the structure to be pushed away, neither during development nor during regeneration (otherwise the morphogen landscape would not be recreated). When a proliferation is triggered by a driver embedded in the existing structure, the new cells are placed in the grid only if the relevant positions are free, so that no old cell is deleted. The proposed mechanisms ensure that the drivers activated to produce a given body part during development are reactivated during regeneration, leading to the same sequence of events (Fig. 4).

### Results and Discussion: Self-generation and self-regeneration of multicellular structures in Epigenetic Tracking

We have run 10 independent simulations of evolution using a genetic algorithm, with constant population size (124 individuals). At the first generation, all genomes were random. At each subsequent generation the genomes of

the individuals in the new population were created as follows: (i) the genomes for the 16 best individual in the previous population were copied from the new population without change (elitism); (ii) 96 genomes were inherited from the previous population with selection probability proportional to each individual's fitness, crossover (one point, with 50% probability) and mutation (the rate is 0.005 per character in the genome); (iii) 12 genomes were created completely randomly. This influx of random genomes introduces new genes into the population to increase evolvability. Another measure to increase evolvability in Epigenetic Tracking, called germline penetration, creates genes with mobile sequences that match mobile codes in the driver cells which have not been activated during development, and inserts them into the genome of next generation's individuals (see Fontana and Wróbel, for the discussion of the biological motivation for this mechanism).

The fitness function rewarded the proximity of the adult structure (the structure after 12 developmental stages) to the shape of a lizard, requiring about 500 000 cells. Each simulation was run for 40 000 generations, and in all simulations the best individual was very close to the target (Fig. 5; only two champions representative for 10 are shown; all the champions had genomes with 80-100 developmental genes). Then, portions of the final structure were removed and allowed to regenerate (Fig. 5) following the debris removal. We performed tens of such experiments, with different damages for different champions, and in all simulations the regeneration was perfect.

Our model of regeneration draws the inspiration from the amazing properties displayed by many biological organisms. The regenerative capabilities in the living world occupy a

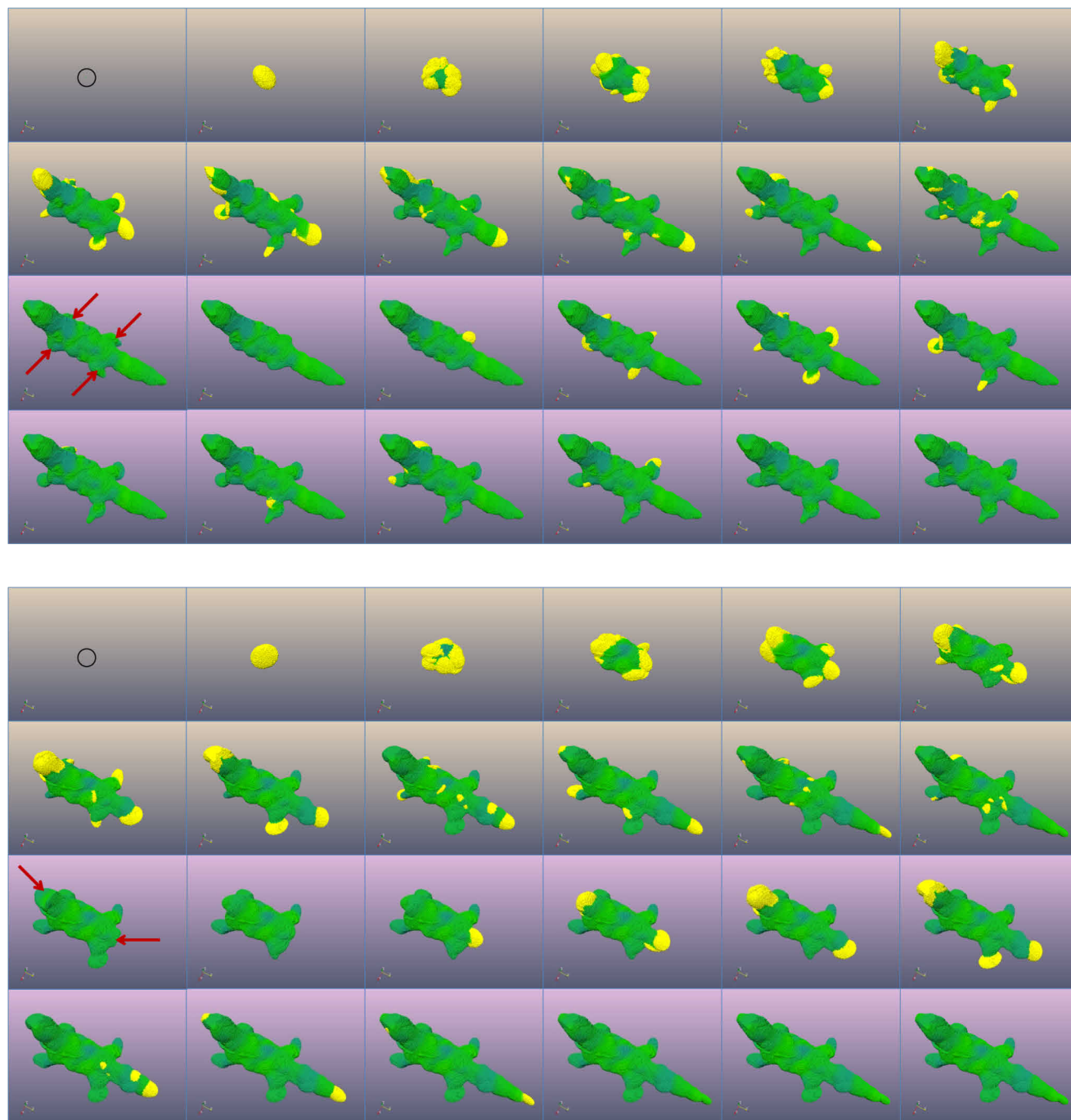


Figure 5: Self-generation and self-regeneration of multicellular structures consisting of hundreds of thousands of cells: the main contribution of this paper. Two champions of independent evolutionary runs are shown in two separate panels, each with 24 frames. In both cases, development unfolds from a single cell (circles) in 12 developmental stages (frames 1-12). After stage 12, the four limbs (top panel) or the tail and the head (bottom panel) are cut off (arrows). After the debris is removed, the structures are completely regenerated.



wide spectrum, ranging from limited cell renewal within tissues, to full regeneration of entire multicellular organisms starting from small fragments. The results obtained in this paper correspond to the latter, most complex extreme of this spectrum, perhaps raising doubts about the biological plausibility of the mechanisms we proposed.

We have built our model of regeneration on two foundations: (i) drivers who were active during development persist, deactivated, in their original positions at later stages of life; (ii) morphogens released from drivers created during later development stages than the damaged part are excluded during regeneration. Both foundations are required to recreate the same conditions during regeneration as the conditions during development in order to guarantee perfect regrowth after damage. A less perfect regrowth – more common in nature – will result, we suspect, if the mechanisms we introduced in Epigenetic Tracking do not work perfectly. We predict that then the regeneration will work better for more elongated parts of the structure (such as limbs or a tail), corresponding to more “isolated” driver subsets, i.e. with less dependencies on subsets in other body parts. More central body parts (e.g., the trunk) correspond to driver subsets with more dependencies, so they require a profound debris removal. We plan to investigate in our future work if these intuitions agree with simulation results for imperfect mechanisms in Epigenetic Tracking.

Are both foundations described above biologically plausible? Drivers in Epigenetic Tracking are inspired by biological embryonic stem cells (Fontana and Wróbel). Embryonic stem cells are totipotent cells, able to differentiate into all cellular types, while adult stem cells are pluripotent cells persisting throughout life, dividing when there is a need to replenish died cells or to regenerate damaged tissues. So persisting quiescent drivers in Epigenetic Tracking can be compared to adult stem cells. On the other hand, perfect debris removal and recreation of the chemical landscape present during development is not entirely plausible, and the fact that it is not may explain why biological organisms with high complexity have a limited ability to regenerate whole parts of their body. Regardless of the degree of biological plausibility, the mechanisms we introduced for simulated development in Epigenetic Tracking could be implemented in artificial physical systems, provided that physical building-blocks able to store genetic information are available.

## Conclusions

Our model of multicellular development, Epigenetic Tracking, allows to self-generate 3-dimensional structures consisting of millions of cells, structures with shapes that have a level of detail unmatched by other models of artificial development. We presented in this paper a new version of Epigenetic Tracking, in which the drivers – cells that correspond to biological organisms, fewer in number than other cells

in the structure – are created using diffusing morphogens, persist through life, and can orchestrate regrowth after damage. The presence of the new mechanisms in the model did not impair evolvability, and allowed for perfect regeneration after damage. We plan to investigate in future work if the regeneration will be less perfect – like in highly complex biological organisms – if these mechanism do not work perfectly. These future work will aim to infer general rules for regeneration in biological and artificial systems.

## References

- De Garis, H. (1999). *Artificial embryology and cellular differentiation*. Academic Press.
- Eggenberger Hotz, P. (2003). Exploring regenerative mechanisms found in flatworms by artificial evolutionary techniques using genetic regulatory networks. In *Proceeding of the Congress on Evolutionary Computation (CEC '03)*, volume 2, pages 2026–2033.
- Fontana, A. (2008). Epigenetic tracking, a method to generate arbitrary shapes by using evo-devo techniques. In *Proceedings of the 8th International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems (EPIROB '08)*.
- Fontana, A. (2009). Epigenetic tracking: biological implications. In *Proceedings of 10th European Conference on Artificial Life (ECAL 2009)*, volume 5777 of LNCS, pages 10–17.
- Fontana, A. (2010). A hypothesis on the role of transposons. *Biosystems*, 101:187–193.
- Fontana, A. and Wróbel, B. A model of evolution of development based on germline penetration of new “junk” DNA.
- Fontana, A. and Wróbel, B. (2013). Morphogen-based self-generation of evolving artificial multicellular structures with millions of cells. *Proceedings of the Annual Conference on Genetic and Evolutionary Computation (GECCO 2013)*, in press.
- Gruau, F., Whitley, D., and Pyeatt, L. (1996). A comparison between cellular encoding and direct encoding for genetic neural networks. In *Genetic Programming 1996: Proceedings of the first annual conference*, pages 81–89.
- Hornby, G. S. and Pollack, J. B. (2002). Creating high-level components with a generative representation for body-brain evolution. *Artificial Life*, 8:223–246.
- Joachimczak, M. and Wrobel, B. (2008). Evo-devo in silico: a model of a gene network regulating multicellular development in 3d space with artificial physics. In



*Proceedings of the 11th international conference on the simulation and synthesis of living systems (ALife XI)*, pages 297–304.

Lindenmayer, A. (1968). Mathematical models for cellular interaction in development. *Journal of Theoretical Biology*, 18:280–289.

Miller, J. F. and Banzhaf, W. (2003). Evolving the program for a cell: from french flags to boolean circuits. In *On growth, form and computers*. Academic Press.

Turing, A. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society*, 237:37–72.

Wolpert, L. and Ticke, C. (2010). *Principles of development*. Oxford University Press.

# Controlling Ant-Based Construction

Lenka Pitonakova<sup>1,2</sup> and Seth Bullock<sup>1</sup>

<sup>1</sup>Institute for Complex Systems Simulation  
University of Southampton, United Kingdom

<sup>2</sup>l.pitonakova@soton.ac.uk

## Abstract

This paper investigates the dynamics of decentralised nest construction in the ant species *Leptothorax tuberointerruptus*, exploring the contribution of, and interaction between, a pheromone building template and a physical building template (the bodies of the ants themselves). We present a continuous-space model of ant behaviour capable of generating ant-like nest structures, the integrity and shapes of which are non-trivially determined by choice of parameters and the building template(s) employed. We go on to demonstrate that the same behavioural algorithm is capable of generating a somewhat wider range of architectural forms, and discuss its limitations and potential extensions.

## Introduction

When building their nests, insect colonies are capable of creating extremely complex structures without employing explicit blueprints (Theraulaz et al., 1998). Such feats of collective construction are achieved via stigmergy, where deposition of building material and pheromones attract, guide and stimulate other nest mates. For example, some ant species progressively encircle their brood with a wall constructed from collected stones (Franks et al., 1992), while paper wasps build structured combs that are sometimes protected by an external envelope (Jeanne and Bouwma, 2002), and termites are capable of building highly complex nests with ventilation shafts, galleries, brood chambers, fungus gardens and royal chambers (Bonabeau et al., 1998; Ladley and Bullock, 2004, 2005).

Due to the inherent parallelism and stochasticity of self-organisation, insect nest construction is prone to problems of interference and needless redundancies are often created (Di Marzo Serugendo et al., 2011). Nevertheless, the fact that the individuals themselves only require limited sensors, memory and reasoning (Mason, 2002) makes attempting to reproduce their behaviour in robots attractive (Holland and Melhuish, 1999; Parker and Zhang, 2006; Bullock et al., 2012). In the future, we might be able to rely on swarms of extremely simple and cheap robots to autonomously build structures guided by environmental cues. For example, we could place signal beacons to suggest where corners of a

building should be or where space should be created for windows and doors.

However, in order to be able to control insect-like construction, we must first understand it. This paper investigates nest building by the ant *Leptothorax tuberointerruptus*, which creates circular structures with one or more entrances around its brood. These nests are created inside flat horizontal cavities and can thus be studied in two dimensions (Franks et al., 1992; Franks and Deneubourg, 1997; Theraulaz et al., 2003). Ant builders can be divided into ‘internal’ and ‘external’ types. Internal ants stay in close proximity to the central brood cluster and tend to push stones away from it, while external ants repeatedly search for stones in the environment and push them directly towards the brood cluster until they collide with another ant or a wall.

The brood cluster and the internal ants that surround it serve as a *physical template* for construction, passively preventing stones from being pushed close to the brood cluster and actively moving stones away from it. Eventually, the built structure itself becomes more important for stigmergy and new stones are often bulldozed into or along existing walls. Building occurs in parallel in several places at once, with some stones travelling between building sites as different ants pick them up and drop them.

There is a certain ambiguity in the literature concerning the role of a *pheromone template* that emanates from the brood cluster. It is clear that ants use it to orient themselves within the nest (Franks et al., 1992), but it is not yet empirically established whether pheromone influences stone deposition directly (Franks and Deneubourg, 1997) leading existing models to differ in how they treat this aspect of ant behaviour (Franks et al., 1992; Theraulaz et al., 2003).

Furthermore, while global colony behaviour is well described (Franks et al., 1992) and modelled analytically (Franks and Deneubourg, 1997), existing agent-based simulations either use grid worlds where noisy movement and bulldozing with friction are not modelled (e.g. Franks et al., 1992) or implement simple continuous behaviour where ants are only of one type (e.g. Theraulaz et al., 2003). Moreover, Theraulaz et al. (2003) also model collisions abstractly, ide-

alising ant stone-dropping behaviour as influenced directly by the local density of stones as well as local pheromone concentration.

Here, we attempt to achieve a better understanding of how the radius and integrity of the constructed nest result from the physical interactions between ants, stones and pheromone in a continuous-space model, where some of the shortcomings of the previous models are addressed. In particular, the following hypotheses are tested:

1. Nests will be larger when there are more internal ants in the colony and smaller when there are more external ants.
2. Using pheromone as a building template will lead to more regular structures, but will also interfere with (1), above.
3. Nest entrances will form spontaneously, influenced by the density of the internal ants and facilitated by the pheromone template.

We also explore extensibility of the building behaviour:

4. It is possible for the same behavioural algorithm to generate alternative architectures by combining multiple pheromone clouds.

## Methods

All simulations were performed in a two-dimensional continuous-space arena of  $660 \times 660$  units ( $165\text{mm}^2$ , i.e. 1 unit =  $0.25\text{mm}$ ). Objects were scaled proportionate to their real-world counterparts using dimensions given by Franks and Deneubourg (1997). At the beginning of each run,  $3 \times 10^3$  rectangular stones of size  $2 \times 2$  units ( $0.5\text{mm}^2$ ) were placed randomly in the arena. The brood was represented by a tight cluster of 50 randomly oriented static ant agents placed around the middle of the arena in a random Gaussian fashion. A circular pheromone cloud 300 units in diameter was centred on the brood cluster such that the pheromone concentration had a constant value of unity at the centre, linearly decreasing to zero at the edge of the cloud. Time was modelled in discrete 0.02-second timesteps and each simulation run lasted 6000 seconds. All results presented in this paper are based on 20 runs per plotted data point.

Initially, a number,  $N_i$ , of ‘internal’ and,  $N_e$ , ‘external’ 10 units  $\times$  2 units ( $2.5\text{mm} \times 0.5\text{mm}$ ) ants were placed around the brood cluster. Ant location, orientation and movement were simulated as continuous (Bourg and Seemann, 2004, p.16–19), where the centre of an ant’s body was moved by a real-valued distance and given a real-valued orientation each time step. Collision detection prevented ants from moving over or through stones or other ants including brood members.

Modelled ants implemented the following empirically observed behaviours (Franks et al., 1992) by executing the algorithm represented in Figures 1–3:

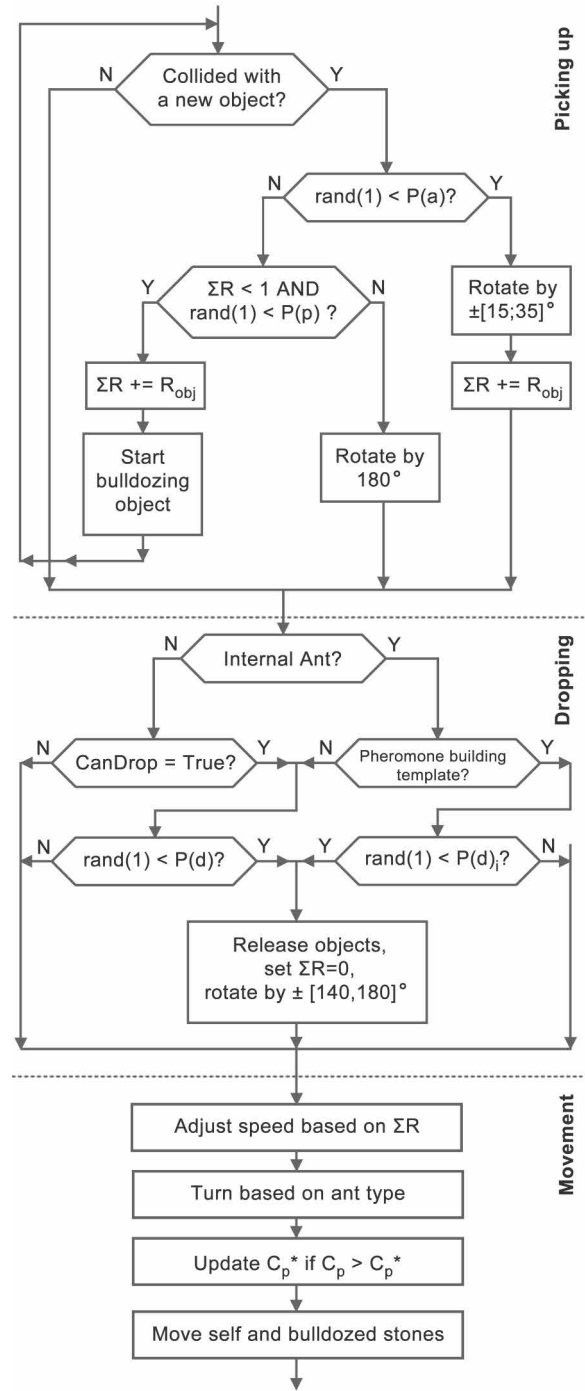


Figure 1: Ant behavioral cycle

1. Random movement, unless:

- bulldozing towards the brood cluster (external ants)
- bulldozing away from the brood cluster (internal ants)
- moving towards the brood cluster when pheromone levels were below a threshold (internal ants only)

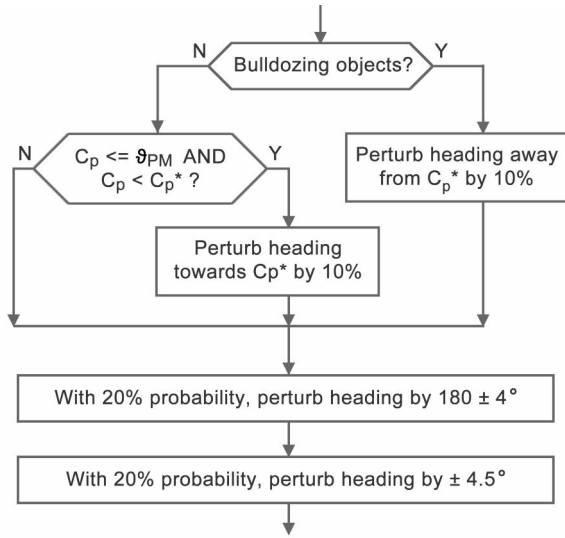


Figure 2: Internal ant's 'Turn based on ant type' routine

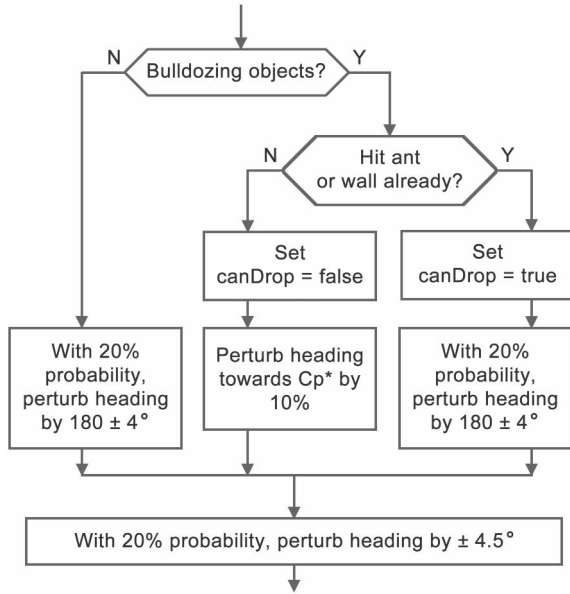


Figure 3: External ant's 'Turn based on ant type' routine

2. Stone bulldozing, i.e. pushing one or more stones forward
3. Stone dropping, the probability of which increased with felt resistance of whatever was being pushed or collided with
4. Occasional moving along walls while bulldozing. In this case, an ant rotated by a small value as it approached an existing wall and continued its movement along it.

In order to orient either towards or away from the brood cluster, both ant types relied on the ability to sense local

pheromone concentration,  $C_p$ , and on remembering the location of the highest pheromone concentration that they had encountered so far,  $C_{p*}$ : a proxy for the location of the brood cluster.

Internal ants (Figure 2) employed a pheromone movement threshold,  $\vartheta_{PM}$ , in order to remain within a characteristic distance of the central brood cluster. Where  $\vartheta_{PM} = 0$ , internal ants were free to roam to the edge of the pheromone cloud. Where  $\vartheta_{PM} = 1$ , internal ants attempted to remain as close to the brood cluster as possible. Internal ants tended to bulldoze encountered stones away from the brood cluster and then returned back. External ants (Figure 3) moved randomly unless they were bulldozing stones towards the brood cluster.

The probability of picking up stones when encountered was a constant  $P(p) = 0.5$ . Each bulldozed stone or stone that an ant was currently colliding with added resistance  $R_s = 0.15$  to the ant's total felt resistance  $\Sigma R \in [0, 1]$ . The resistance experienced during collisions with other ants was  $R_a = 1.0$ .

When an ant pushed more stones, its speed decreased (Equation 1), while its probability to move along walls  $P(a)$  and to drop stones  $P(d)$  increased (Equations 2 and 3), causing the colony to gradually extend existing walls rather than to make them thicker. Note that external bulldozers only started checking whether they should drop stones after they encountered an immovable obstacle and turned away from it, while internal bulldozers could drop stones at any time.

$$speed = 2 \times (1 - \Sigma R) \quad (1)$$

$$P(a) = \Sigma R \quad (2)$$

$$P(d) = f \times |\log(1 - \alpha \times (\Sigma R + \epsilon))|; \quad (3)$$

$$f = 0.625, \alpha = 0.8, \epsilon = 10^{-11}$$

In runs where the pheromone building template was employed, local pheromone concentration,  $C_p$ , influenced each internal ant's drop probability,  $P(d)_i$ , such that it exponentially increased as the ant moved towards edges of the pheromone cloud (Equation 4).

$$P(d)_i = \min(1, P(d) + |g \times \log(C_p)|); \quad (4)$$

$$g = 1/7$$

The combined effect of  $\Sigma R$  and  $C_p$  on  $P(d)$  and  $P(d)_i$  is depicted in Figure 4.

## Results

Structures built by the artificial ants were generally circular (Figure 5), with (sometimes incomplete) walls forming around the brood cluster at a characteristic distance that varied with model parameters. This result was robust with respect to the colony size and suitable values of the pheromone movement threshold,  $\vartheta_{PM}$ , and is comparable with the real and simulated ants in the existing literature (Franks et al., 1992; Theraulaz et al., 2003).

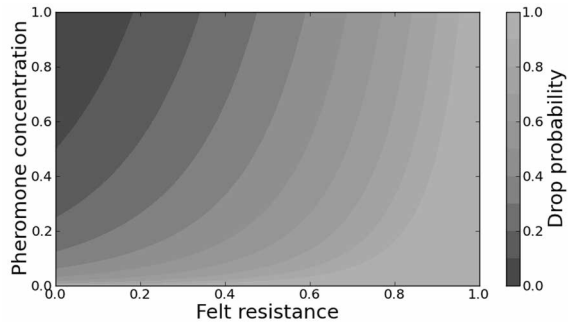


Figure 4: Contour plot of drop probability  $P(d)_i$  as a function of felt resistance,  $\Sigma R$ , and local pheromone concentration,  $C_p$ , based on Equations 3 and 4. In the absence of the pheromone building template, values of  $P(d)$  are dependant on Equation 3 alone and are plotted at  $y = 1$ , i.e. where the log of pheromone concentration is zero.

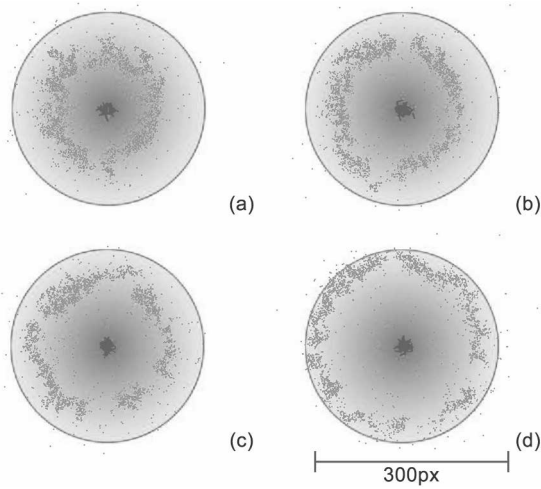


Figure 5: Example nests for various ant parameter combinations: (a)  $N_i=10$ ,  $\vartheta_{PM}=1.0$ ; (b)  $N_i=30$ ,  $\vartheta_{PM}=0.75$ ; (c)  $N_i=50$ ,  $\vartheta_{PM}=1.0$ ; (d)  $N_i=50$ ,  $\vartheta_{PM}=0.5$ . The pheromone cloud is shown as gray gradient. Brood clusters placed in the arena centres are shown in dark gray.

To check for the influence of stigmergy, we explored changes in the mean length of each bulldozing episode for external ants. As expected, the mean bulldozing duration, measured from when an external ant bulldozed into the pheromone cloud to the terminal collision, decreased over time irrespective of values of  $\vartheta_{PM}$  (Figure 6) due to the progressively higher frequency of encountering already placed stones. This behaviour is consistent with Franks and Deneubourg (1997), who implied that stone carrying time decays exponentially as the nest building progresses.

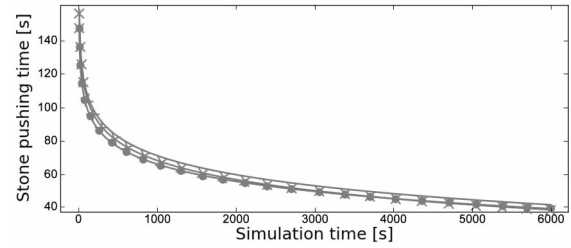


Figure 6: Fitted exponential models of mean bulldozing time measured for external ants between the moment they entered the pheromone cloud and the moment they dropped stones.  $N_i=30$ ,  $N_e=10$  and  $\vartheta_{PM}=1.0$  (solid,  $R^2 = 39.97\%$ ),  $\vartheta_{PM}=0.75$  (crosses,  $R^2 = 37.83\%$ ),  $\vartheta_{PM}=0.5$  (dots,  $R^2 = 28.04\%$ ). (Note that the inherent stochasticity of the simulation means that the exponential model does not account for all of the variance in bulldozing times.)

### Physical Building Template Only

We first explored building behavior in the absence of a pheromone building template.

The size of nests built by real ant colonies depended on the number of colony members (Franks and Deneubourg, 1997). Similarly, increasing the number of simulated internal ants,  $N_i$ , tended to increase the effective diameter of final structures (Figure 7a). Increasing colony size also increased the irregularity of the built structures, measured as the standard deviation of the number of stones found in eight regular conical sectors, each originating from the centre of the pheromone cloud (Figure 7b).

Both nest diameter and nest regularity were also dependent on the value of  $\vartheta_{PM}$ , i.e. on how far from the brood cluster internal ants were ‘willing’ to roam before turning back towards it. Only when  $\vartheta_{PM}$  was higher than a specific threshold value, was a colony able to effectively encircle itself with stones.

Where  $\vartheta_{PM}$  was high, internal ants were tightly clustered and regular circular nests were built relatively close to the central brood cluster. For runs with lower  $\vartheta_{PM}$ , internal ants spread out from the brood cluster, pushing the built structure out, but their lower density at the characteristic radius of the nest wall allowed gaps to form, compromising its regularity. For even lower values of  $\vartheta_{PM}$ , the internal ants were spread out to such an extent that they cease to be an effective physical template for building, allowing external ants to build much closer to the brood cluster. The critical value of  $\vartheta_{PM}$  (below which the physical building template fails) varied inversely with  $N_i$  as larger colonies could achieve sufficient density at greater distance from the brood cluster.

As  $\vartheta_{PM}$  approached zero and the density of internal ants was minimised, many stones were left very near the middle of the pheromone cloud, since external ants were often able to bring them to the brood and there was a lower probability



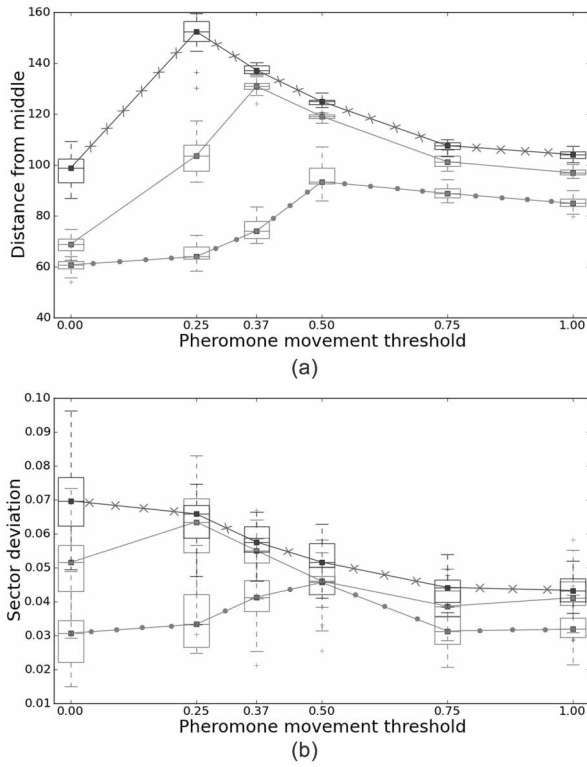


Figure 7: Influence of  $\vartheta_{PM}$  on (a) average distance of stones from the middle of the pheromone cloud, and (b) standard deviation of the number of stones in eight regular conical sectors of the cloud for  $N_e=10$  (all) and  $N_i=10$  (dots),  $N_i=30$  (solid),  $N_i=50$  (crosses).

of an internal ant encountering and removing them. In these cases, built structures were packed close to the brood and were often very irregular.

The number of external ants also had a non-trivial influence on building behaviour. With the number of internal ants fixed at  $N_i = 30$  we explored the influence of increasing external ant numbers from 10 to 180 for two pheromone thresholds,  $\vartheta_{PM} = 0.5$  and  $\vartheta_{PM}=1.0$ . Increasing the number of external ants,  $N_e$ , from 10 to 30 caused nests to become smaller, as expected, since increased pressure from external ants tended to establish walls closer to the brood cluster (Figure 8a). However, contra to Hypothesis 1, further increases to  $N_e$  gave rise to larger nests as a consequence of interference between external ants, some of which found themselves within the pheromone cloud during or after bulldozing, effectively augmenting the number of internal ants and amplifying the effect of their physical building template. Increasing the number of external ants,  $N_e$ , also tended to increase the amplification of their initial building sites, decreasing the regularity of built structures (Figure 8b).

In summary, this section has confirmed that model ants are able to achieve built structures the size and integrity of

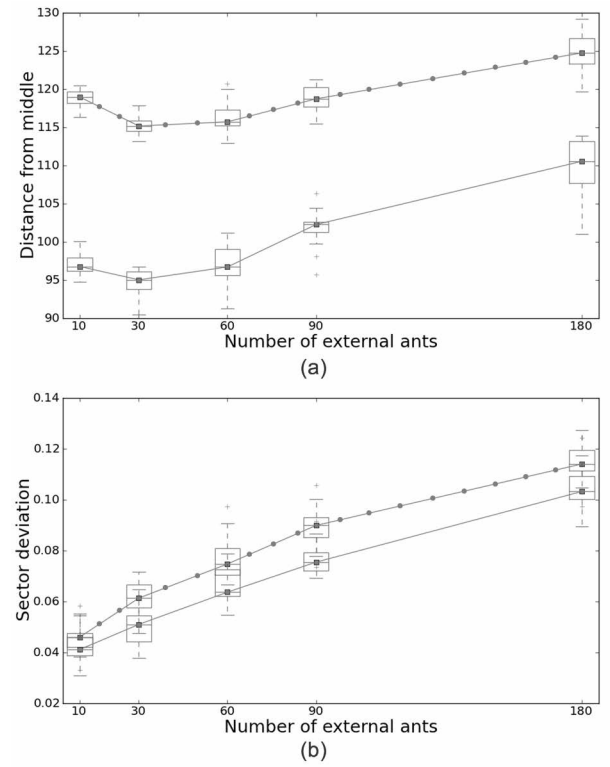


Figure 8: Influence of external ant number on (a) average distance of stones from the middle of the pheromone cloud, and (b) standard deviation of the number of stones in eight regular conical sectors of the cloud for  $N_i=30$  (all) and  $\vartheta_{PM}=0.5$  (dots),  $\vartheta_{PM}=1.0$  (solid).

which reflect a complicated interaction between the size of a colony and the extent to which internal ants tend to wander from the brood. They are able to achieve these structures in the absence of a pheromone building template, i.e. pheromone-mediated dropping behaviour is not necessary for successful nest formation. In the next section we explore the influence of such a pheromone building template and assess entrance formation.

### Physical+Pheromone Building Templates

The effect of employing a pheromone building template to encourage the dropping behaviour of internal ants (Equation 4) is depicted in Figure 9. Ants mostly tended to build smaller and more regular structures as dropping became more precisely tied to an internal ant's location within the pheromone cloud. This effect was more significant for larger colonies and those where internal ants roamed further from the brood cluster.

When the pheromone building template was used with large numbers of internal ants, e.g.  $N_i = 50$ , the influence of pheromone on building initially caused stones to be dropped closer to the brood cluster, but nests were subsequently ex-

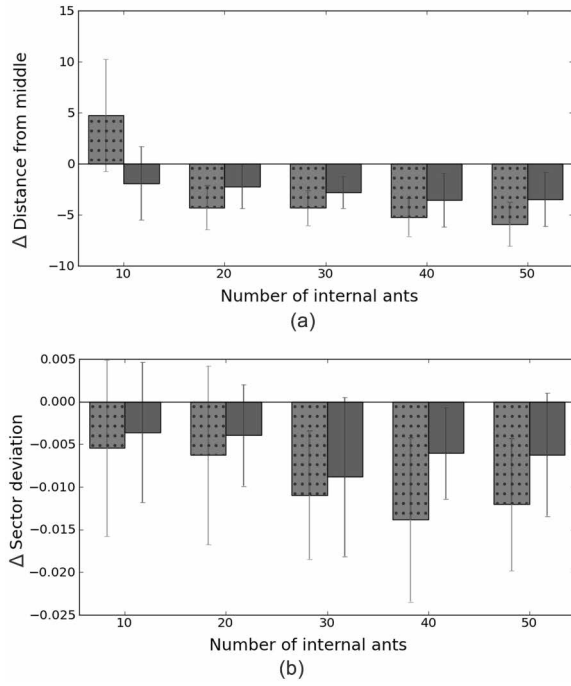


Figure 9: Difference between runs with and without pheromone building template in terms of (a) average distance of stones from the middle of the cloud where a negative number indicates smaller nests with pheromone building templates, and (b) standard deviation of the number of stones in eight conical sectors of the cloud where a negative number indicates more regular structures with the pheromone building templates, using  $N_e=10$  with  $\vartheta_{PM}=0.5$  (dots) and  $\vartheta_{PM}=1.0$  (solid).

panded to some extent due to the pressure of internal ants within the walls.

By contrast, when the number of internal ants was small, e.g.  $N_i = 10$ , and they remained close to the brood cluster ( $\vartheta_{PM} = 1.0$ ), the pheromone building template had little effect since internal ants rarely roamed a significant distance from the brood cluster.

However, when the same small number of internal ants were allowed to roam further from the brood cluster ( $\vartheta_{PM} = 0.5$ ), whereas without a pheromone building template they failed to achieve a nest wall, tending to assemble many stones in the middle of the cloud, with a pheromone building template more regular structures were achieved at an increased average distance from the brood cluster. In this case, the pheromone building template made nest creation possible when it otherwise would not be, by encouraging a wall to be built closer to the brood.

### Entrance Formation

Experimental runs were evaluated manually in order to categorise the final structures by number of entrances and

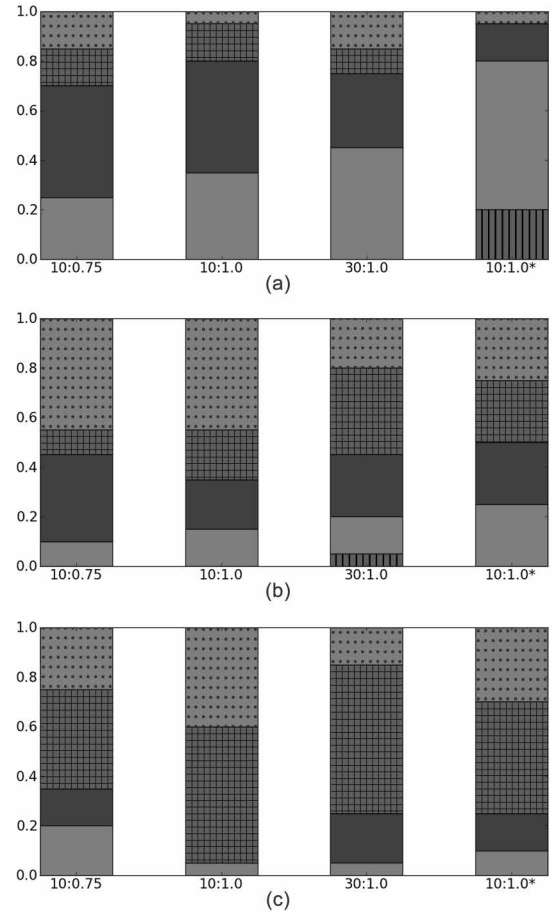


Figure 10: Proportion of nests with 0 (bars), 1 or 2 (light solid), 3 (dark solid) or more entrances (hatch) and irregular structures (dots) using (a)  $N_i=10$ , (b)  $N_i=30$  and (c)  $N_i=50$ . The individual groups are labeled using pattern  $N_e : \vartheta_{PM}$ . A star (\*) indicates that the pheromone building template was employed.

whether they could be considered nests at all (Figure 10). The most regular nests were built when 10 internal ants were used. Irregular nests (i.e. arrangements of stones that did not form a coherent structure at all) occurred only 5% of the time when  $\vartheta_{PM}=1.0$  and 15% of the time when  $\vartheta_{PM}=0.75$ . The frequency of nests with only 1 or 2 entrances increased as the number of external ants increased and similarly when the pheromone building template was used. However, in the latter case, the ants also built a complete wall with no entrances at all in 4/20 runs.

A similar pattern of entrance formation was observed in colonies with 30 internal ants, although generally the frequency of irregular structures increased in comparison with the previous case. Furthermore, these larger colonies tended to build nests with three or more entrances more often, especially when  $N_e = 30$  (12/20 runs). A nest with no entrances

was only built on one occasion when  $N_e=30$ .

The trend to create more entrances was even stronger for colonies with 50 internal ants. This was especially true when  $\vartheta_{PM}=1.0$ , in which case nests with four or more entrances formed in 11/20 runs. Interestingly, regularity of nests increased in comparison with colonies where  $N_i = 30$  ( $N_i = 30$  and  $\vartheta_{PM} = 0.75$  or  $\vartheta_{PM} = 1.0$ , regular nests occurred in 11/20 runs, while  $N_i = 50$ ,  $\vartheta_{PM} = 0.75$ , in 15/20 runs and  $N_i = 50$ ,  $\vartheta_{PM} = 1.0$ , in 12/20 runs). Nest regularity increased slightly in the experiments with pheromone building templates ( $N_e = 10$ ,  $\vartheta_{PM} = 1.0$  regular nests in 12/20 runs and in 14/20 runs during the pheromone building template experiments), although regular nests were the most frequent when  $N_e = 30$  and  $\vartheta_{PM} = 1.0$  (16/20 runs).

### Controlling Nest Shape

Standard structures created by ants with a single pheromone cloud were circular. In the following set of experiments, the pheromone cloud diameter was decreased from 300 units to 150 units and a number of clouds with brood clusters at their centres were arranged in order to create nests of different shapes.

Three experimental setups were created: a) rectangle: two clouds were horizontally aligned and the distance of their centres was set to 75 units, b) triangle: three clouds, the centers of which formed the vertices of a triangle with sides 75 units long, and c) square: four clouds, the centers of which formed corners of a square with 75 units side length. The number of external ants was 10, while there were always 10 internal ants per pheromone cloud. The value of  $\vartheta_{PM}$  was set to 1.0, since the previous experiments showed that the most regular nests were built with this value (Figures 7–10). Note that ants were unable to distinguish amongst the different sources of pheromone.

The final positions from 20 runs in each experiment were amalgamated in order to generate contour plots (Figure 11). In each experiment, the desired shape was always achieved, although there were no sharp corners as walls naturally curved around the boundaries of the individual pheromone clouds. Once again, using the pheromone building template facilitated creation of more regular structures, although it was not required to achieve the desired shapes.

One or two entrances usually formed along the shorter edges of rectangular structures. The triangular nests had one to three small entrances that could be found near the vertices. Entrances in square nests were usually more numerous and formed both along the edges and in the corners. Probably due to their size, square structures had the least regular distribution of stones in their walls.

### Discussion

Nest formation by simulated ants was tested in a number of different scenarios. The final circular structures, as well as the process by which they were built were comparable to real

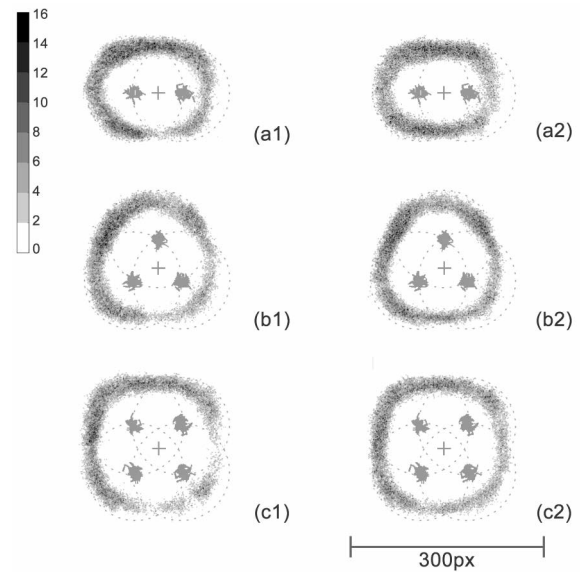


Figure 11: Contour plots of nests created during the (a) rectangle, (b) triangle, and (c) square experiments. Results from experiments (1) without a pheromone building template, and (2) with such a template are shown. Pheromone clouds are represented by dotted circles. Crosses show arena centres, with brood clusters around them.

and previously simulated ants (Franks et al., 1992; Theraulaz et al., 2003) across a wide parameter space. Usually, the ants initially created a number of stone heaps that were gradually extended and connected together, while nest entrances remained clear throughout the process. Adding more external ants initially caused gaps in the nest wall to be created and destroyed, with stable entrances appearing only later in the simulations.

This building behaviour was more similar to that of real ants that clear a cavity of stones and create a number of progressively joined heaps rather than of those that bring stones from outside of the building site and gradually form a C-shaped nest with only one entrance (Franks et al., 1992). It is possible that in the latter case, external ants carry stones towards the nest from one direction, rather than from all directions as was the case in the simulated arena, or that they find stones further away from the nest, causing a slower stone intake rate and thus different wall formation dynamics.

The nest size varied as the number of internal ants increased, confirming the assumption of Hypothesis 1, although regularity of the structures decreased when they became large. The differences in nest size occurred due to variations in ant movement as large colonies required more space to spread out. Similarly, larger structures were formed when movement of ants within the pheromone cloud became less restricted by varying the  $\vartheta_{PM}$  parameter.

On the other hand, the assumption of Hypothesis 1 that

larger numbers of external ants  $N_e$  would cause higher pressure and lead to the creation of smaller nests was only true when  $N_e$  was relatively low. The nests actually became larger and less regular when  $N_e \geq 90$ . This surprising increase in nest diameter occurred because more external ants could be found inside the pheromone cloud, adding to the size of the physical template formed by the internal ants and brood. Nest regularity decreased as higher numbers of external ants assembled stones more rapidly, increasing the probability of amplifying any initial building sites.

Use of the pheromone cloud as a template for building improved nest regularity (Figures 9, 10, 11). Furthermore, the resulting structures were smaller as the gradient of pheromone concentration interfered with the effect of ant movement, confirming Hypothesis 2. This effect could not be observed with only 10 internal ants when they were restricted to remain very close to the centre of the pheromone cloud as they could not reach places influenced by the pheromone building template.

The fact that the pheromone template was not required for successful nest construction agrees with the assumption of Franks and Deneubourg (1997) who understood pheromone as simply a cue for ants to orient themselves within the nest. Allowing pheromone to influence stone deposition directly (e.g. Theraulaz et al., 2003) thus seems unnecessary.

The assumption of Hypothesis 3, that nest entrances would form when internal ant movement is more constrained ( $\vartheta_{PM}$  is high), was partially supported (Figure 10). When the number of internal ants was low,  $N_i = 10$ , more regular structures were produced for  $\vartheta_{PM} = 1.0$  compared to  $\vartheta_{PM} = 0.75$ . However, the effect of  $\vartheta_{PM}$  was not apparent when  $N_i = 30$  and was reversed when  $N_i = 50$ , i.e., the influence of  $\vartheta_{PM}$  on nest regularity varied with colony size. On the other hand, use of the pheromone building template always improved nest regularity, as predicted.

Finally, it was shown that non-circular nest shapes can be created when multiple pheromone clouds are arranged together (Figure 11), as predicted by Hypothesis 4. The clouds needed to be small enough so that there were enough stones available to create the final shapes and also suitably close to each other so that internal ants could travel between them.

## Conclusion

In conclusion, this work has helped to understand the building behaviour of the ant *Leptothorax tuberointerruptus* and to answer questions about the roles of colony size and pheromone-mediated behaviour in the building process. It is clear from the results presented here that even in the very idealised and simple scenario that we explore, the interactions that give rise to built structures are subtle and complex. We also show that this simple behavioural algorithm could perhaps be applied with cheap robotic ants to create structures beyond the circles achieved by *L. tuberointerruptus*. Extensions to the model include adding agent-generated

pheromone gradients, and applying the revealed principles of nest morphogenesis to the decentralised construction of more complicated heterogeneous architectures.

**Acknowledgements:** This work was supported by an EPSRC Doctoral Training Centre grant (EP/G03690X/1).

## References

- Bonabeau, E., Theraulaz, G., Deneubourg, J.-L., Franks, N. R., Rafelsberger, O., Joly, J.-L., and Blanco, S. (1998). A model of the emergence of pillars, walls and royal chambers in termite nests. *Philosophical Transactions of the Royal Society of London, Series B*, 353:1561–1576.
- Bourg, D. M. and Seemann, G. (2004). *AI for Game Developers*. O'Reilly Media.
- Bullock, S., Ladley, D., and Kerby, M. (2012). Wasps, termites, and waspmites: Distinguishing competence from performance in collective construction. *Artificial Life*, 18(3):267–90.
- Di Marzo Serugendo, G., Gleizes, M.-P., and Karageorgos, A. (2011). Self-organising systems. In Di Marzo Serugendo, G., Gleizes, M.-P., and Karageorgos, A., editors, *Self-organising Software*, Natural Computing Series, pages 7–32. Springer.
- Franks, N. and Deneubourg, J. (1997). Self-organizing nest construction in ants: Individual worker behaviour and the nest's dynamics. *Animal Behaviour*, 54(4):779–96.
- Franks, N., Wilby, A., Silverman, B., and Tofts, C. (1992). Self-organizing nest construction in ants: Sophisticated building by blind bulldozing. *Animal Behaviour*, 44:357–375.
- Holland, O. and Melhuish, C. (1999). Sorting in collective robotics. In Langton, C. G. and Shimohara, T., editors, *Artificial Life V*, pages 173–202. MIT Press.
- Jeanne, R. L. and Bouwma, A. M. (2002). Scaling in nests of a social wasp: A property of the social group. *The Biological Bulletin*, 202(3):289–95.
- Ladley, D. and Bullock, S. (2004). Logistic constraints on 3D termite construction. In Dorigo, M., Birattari, M., Blum, C., Gambardella, L. M., Mondada, F., and Stutzle, T., editors, *The Fourth International Workshop on Ant Colony Optimisation*, pages 178–189. Springer, Berlin.
- Ladley, D. and Bullock, S. (2005). The role of logistic constraints on termite construction of chambers and tunnels. *Journal of Theoretical Biology*, 234:551–564.
- Mason, Z. (2002). Programming with stigmergy: Using swarms for construction. In Standish, R., Bedau, M. A., and Abbass, H. A., editors, *Artificial Life VIII*, pages 371–374. MIT Press.
- Parker, C. A. C. and Zhang, H. (2006). Collective robotic site preparation. *Adaptive Behaviour*, 14(1):5–19.
- Theraulaz, G., Bonabeau, E., and Deneubourg, J.-L. (1998). The origin of nest complexity in social insects. *Complexity*, 3(6):15–25.
- Theraulaz, G., Gautrais, J., Camazine, S., and Deneubourg, J.-L. (2003). The formation of spatial patterns in social insects: from simple behaviours to complex structures. *Philosophical Transactions of the Royal Society of London, Series A*, 361(1807):1263–82.



# The coevolution of costly heterogeneities and cooperation in the prisoner's dilemma game

Markus Brede<sup>1</sup> and Jason Noble<sup>1</sup>

<sup>1</sup>University of Southampton, Southampton, SO17 1BJ  
markus.brede@soton.ac.uk

## Abstract

This paper discusses the co-evolution of social strategies and an efficiency trait in spatial evolutionary games. The continuous efficiency trait determines how well a player can convert gains from a prisoner's dilemma game into evolutionary fitness. It is assumed to come at a cost proportional to its magnitude and this cost is deducted from payoff. We demonstrate that cost ranges exist such that the regime in which cooperation can persist is strongly extended by the co-evolution of efficiencies and strategies. We find that cooperation typically associates with large efficiencies while defection tends to pair with lower efficiencies. The simulations highlight that social dilemma situations in structured populations can be resolved in a natural way: the nature of the dilemma itself leads to differential pressures for efficiency improvement in cooperator and defector populations. Cooperators benefit by larger improvements which allow them to survive even in the face of inferior performance in the social dilemma. Importantly, the mechanism is possible with and without the presence of noise in the evolutionary replication process.

## Introduction

Altruism – acting for the benefit of the group even if costly to the individual – is a widespread phenomenon in social and biological systems. Examples range from simple micro-organisms (Crespi, 2001) up to complex social interactions in society (Beinhocker, 2007). Various aspects of understanding the emergence and sustainability of such behaviour still poses a major challenge to evolutionary game theory (Weibull, 1996) and recent decades have seen very active research in the field (Wang et al., 2012; Szolnoki et al., 2009a; Szolnoki and Szabo, 2004; Tanimoto and Yamauchi, 2010; Masuda, 2007; Abramson and Kuperman, 2001; Tanimoto and Yamauchi, 2012; Santos et al., 2006a; Zimmermann and Eguíluz, 2005; Brede, 2011b; Perc and Wang, 2010; Wang and Perc, 2010; Szolnoki et al., 2009b; Szolnoki and Szabó, 2007; Brede, 2011a; Cao et al., 2011; Zhang et al., 2010; Szolnoki et al., 2010; Szolnoki and Perc, 2008; Szabo and Hauert, 2002; Perc and Szolnoki, 2008; Santos et al., 2006b; Brede, 2013b; Van Segbroeck et al., 2009; Szolnoki et al., 2012; Chadeaux and Helbing, 2010).

Models of the evolution of cooperation often build on the paradigmatic scenario described in the prisoner's dilemma.

In the simple one-off game two players are confronted with a simultaneous choice between two pure strategies, frequently labelled as “C” (for cooperate) and “D” (for defect). Depending on the combinations of choices, payoffs from the game are as follows. Mutual cooperation is rewarded with a payoff of  $R$  for both players, a player who plays “D” against “C” receives the temptation to defect  $T$  while the cooperator is paid the sucker's payoff  $S$  and mutual defection results in a payment of  $P$  for both players. For the prisoner's dilemma the ranking of payoffs is  $T > R > P > S$  and  $2R > T + S$ , such that the optimal choice for an individual who wants to maximize its own game outcome is always “D” while “C” is the optimal choice of a central planner interested in the good of the group.

A common explanation for the sustainability of cooperative strategies assumes positive assortment such that strategies of the same type can interact more often than when population structures are well mixed, cf. e.g., (Eshel and Cavalli-Sforza, 1983; Nowak and M., 1992). Such positive assortment can be facilitated by ‘network reciprocity’ in structured populations (Nowak, 2006; Szabó and Fath, 2007). Especially since the classification of prototypical network structures, like scale-free and small-world type networks, evolutionary game theory in structured populations has found growing interest. An important discovery in this line of research has been that cooperative strategies can receive a strong boost in populations that are coupled by very heterogeneous networks (Santos et al., 2006b), but notice the role of game participation costs in this effect (Masuda, 2007; Tanimoto and Yamauchi, 2010). Later work clarified that also other types of heterogeneity, e.g. in abilities of players to generate payoff (Perc and Szolnoki, 2008; Brede, 2011a) or in differing abilities of players to pass on strategies or adapt to neighbours (Szolnoki and Szabó, 2007; Wang and Perc, 2010; Perc and Wang, 2010; Tanimoto and Yamauchi, 2012), can give similar support for cooperation, even if the network of social interactions is regular.

Some recent studies have started to focus on the question how heterogeneity and game strategies can co-evolve, see (Perc and Szolnoki, 2010) for a review. The most prominent



approach in the field is probably to study adaptive networks in which social interactions change at a timescale similar to that of the evolution of game strategies (Zimmermann and Eguíluz, 2005; Santos et al., 2006a; Van Segbroek et al., 2008; Cao et al., 2011). A crucial assumption in these models is that agents have the cognitive abilities to break off undesirable ties.

Other models have considered the co-evolution of slow and fast strategy pass, e.g. via considering age-dependent abilities of agents (Wang et al., 2012), the co-evolution of performance evaluation rules (Brede, 2013b) or reinforcement of the position of abilities of agents who successfully passed on their strategies in past interactions (Szolnoki and Perc, 2008; Szolnoki et al., 2010; Zhang et al., 2010). As noted in (Brede, 2013a), common to these approaches is an assumption of a dynamics similar to Hebbian learning (Hebb, 1949): Successful interactions become stronger while unsuccessful interactions tend to decline in frequency. Whilst such processes based on Hebbian learning may be reasonable models of social interactions in many contexts, they still rest on ad-hoc assumptions (i.e. those of a Hebbian-like dynamics of system structure, or abilities to break unprofitable links (Van Segbroek et al., 2008) or some mechanism to influence group formation (Powers et al., 2011)) and do not provide a purely evolutionary framework that describes the co-evolution of system structure and social strategies via the same mechanism of evolution. Further, many of these models can only support cooperation if additionally constrained: I.e. in adaptive network models connectivities are typically held constant, or in the ageing-based models maximum ages are imposed, or in the reinforcement model the rate of reinforcement is found to be required to be within a certain range for optimal support of cooperation.

A recent paper addresses this gap and proposes a model in which traits of slow and fast strategy pass of agents can co-evolve with social strategies to support cooperation (Brede, 2013a). The paper proposes a framework in which agents can enhance their abilities to pass on strategies, albeit at a cost. Considering the binary options of ‘advertising’ (i.e. investing in fast strategy spread at a cost) or not advertising (i.e. normal strategy spread at no cost) the study demonstrates that cost-regimes exist, such that cooperation can associate with costly fast strategy spread while this is not viable for defection. It is easy to understand why this is the case: In comparison to defectors cooperators benefit from an investment to surround themselves with like types and thus they can afford to invest more in costly strategy pass than defectors. Hence, if strategy pass is costly enough, the usual competition between cooperators and defectors is replaced by a competition between fast spreading cooperators and slow spreading defectors, resulting in an evolutionary benefit to the former. The model of (Brede, 2013a) considers a binary choice (i.e. advertise or don’t advertise). Further,

the cooperation-supporting dynamics of (Brede, 2013a) relies on the crucial assumption of noise in strategy replication without which the mechanism of costly advertising cannot operate and the model is very sensitive to assumptions about joint inheritance of the advertising and the social strategies.

In this paper we consider a slightly altered modelling framework and illustrate that a co-evolutionary dynamics of agents with to a different extent enhanced abilities to generate payoff from the game can support cooperation even without these key ingredients of (Brede, 2013a), i.e. without the assumption of binary strategies, and without the assumption of noise in strategy replication.

## Model

Consider a spatially distributed population of  $N = L \times L$  agents that interact with their von Neumann neighbours. Agents are characterized by social strategies  $s \in \{C, D\}$  which they employ when playing a one-off prisoner’s dilemma game with their neighbours. The game is parametrized in the conventional way via  $R = 1$ ,  $T = 1 + r$ ,  $S = -r$  and  $P = 0$ . As usual the parameter  $r \in (0, 1)$  characterises the dilemma strength.

Every round an agent  $i$  earns payoff  $\pi$  from interactions with all four spatial neighbours. Further, every agent  $i$  is characterised by a trait  $\epsilon_i$  that determines the efficiency with which it can convert payoff gleaned from the game into evolutionary fitness  $f$  such that

$$f_i = (1 + b\epsilon_i)(\pi_i - c\epsilon_i). \quad (1)$$

The motivation for Eq. (1) is that every agent has a default mechanism to convert payoff into fitness at unit rate (represented by the term  $\pi_i$  in the expansion of (1)). However, after playing the game it can also invest an amount  $c\epsilon_i$  into higher efficiency conversion. Hence, after playing the PD game, a cost  $c\epsilon_i$  is deducted from game payoff and then the remaining payoff is converted into fitness. An alternative model is that the cost of efficiency improvements is deducted after payoff is converted into fitness, resulting in  $f_i = (1 + b\epsilon_i)\pi_i - c\epsilon_i$ . Both models result in qualitatively similar dynamics and we focus on the first choice in this paper.

In our model the trait  $\epsilon_i$  represents the ‘biological machinery’ to make use of payoff from the game,  $b$  measures its efficiency, and  $c$  is the cost (per unit of  $\epsilon$ ) to maintain it. In a biological context the cost of enhanced efficiencies could be seen as a cost to maintain a certain body mass, in a social context it might be associated with the maintenance of equipment or a cost to acquire certain skills. The assumption of a linear relationship between the size of the trait and the cost is for simplicity, in general it might be more reasonable to assume a different monotonic non-linear relationships, but this will not alter qualitative results.

We then carry out evolutionary simulations based on the following protocol:

- The lattice is seeded with random initial conditions, i.e. with probability 1/2 agents are assigned the social strategy  $C$  and with probability 1/2 social strategy  $D$ . Agents are also initialized with efficiency traits  $\epsilon$  selected uniformly at random from the interval  $[0, 1]$ .
- A focus agent  $i$  is picked at random from the population of  $N$  agents and one of its neighbours is selected at random as a reference agent  $j$ .
- Focus' and reference agent's payoffs  $\pi_i$  and  $\pi_j$  are evaluated and converted into evolutionary fitness according to Eq. (1).
- With probability

$$P(j \rightarrow i) = \frac{\exp(f_j/\kappa)}{\exp(f_j/\kappa) + \exp(f_i/\kappa)} \quad (2)$$

the focus agent  $i$  will adopt the reference agent  $j$ 's traits, i.e. the social strategy  $s_j$  and the efficiency trait  $\epsilon_j$  of  $j$ . As introduced in (Szabó and Toke, 1998) the parameter  $\kappa$  in (2) gives the noise level in the process of strategy spread. For  $\kappa = 0$  superior performers always replace worse performers, if  $\kappa > 0$  also less successful strategies have an occasional chance to invade a neighbours place. Note, that the noise parameter is of importance to contrast the present results with those of (Brede, 2013a), because the results of (Brede, 2013a) are not robust in the limit  $\kappa \rightarrow 0$  of (up to neighbour selection) deterministic updating.

- The process of game play and replication is iterated till a quasistationary state is reached and then average frequencies of cooperators and defectors and equilibrium averages over the evolutionary trait are calculated from a sufficient number of further iterations.
- The entire experiment is then repeated a sufficient number of times to evaluate from how many random initial conditions cooperation could evolve.

Numerical results presented below are generally obtained from simulations on  $200 \times 200$  tori and  $b = 2$  and have been repeated for at least 50 times to obtain estimates of the frequency of situations in which cooperation can arise.

## Results

This section describes and analyses numerical results obtained by simulations of the model introduced above. Figure 1 compares average trajectories of the co-evolution of the efficiency trait and social strategies for a case when the efficiency trait is costly and another in which it comes for free. Both scenarios are for the case of noiseless replication  $\kappa = 0$ . Notably, if efficiency is not costly, both cooperators (open boxes) and defectors (filled boxes) evolve to maximize efficiencies. Asymptotically, this results in a

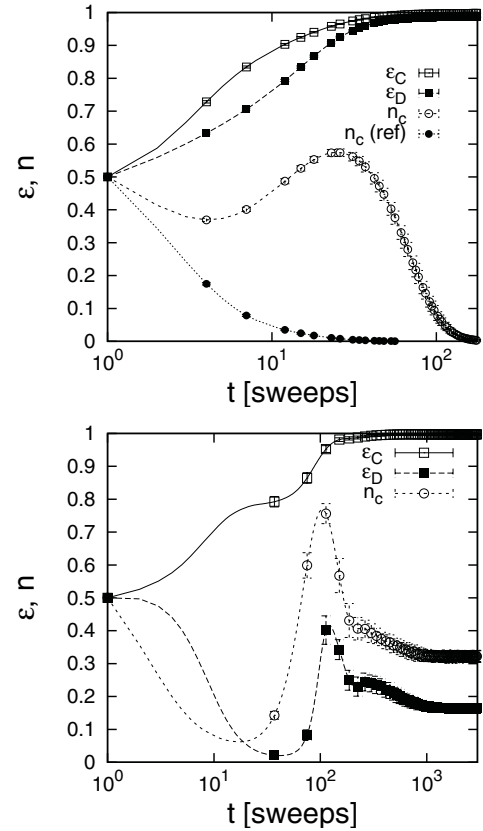


Figure 1: Co-evolution of social strategies and efficiency traits for (top)  $r = 0.01$  and cost  $c = 0$  and (bottom)  $r = 0.1$  and cost  $c = 1$  for  $\kappa = 0$ . Average trajectories for the density of cooperators  $n_c$  and the average efficiency trait of cooperators  $\epsilon_C$  and defectors  $\epsilon_D$  have been calculated from sampling the stochastic dynamics of the evolution over 1000 independent runs on a  $200 \times 200$  torus. For comparison the figure also contains the average evolution of cooperators in the standard one-off spatial game with  $\kappa = 0$  and  $r = 0.01$ . Note the logarithmic scale for the time domain.

homogeneous system in which payoffs are scaled by a factor  $1 + b$  and cooperative strategies cannot survive for even very low dilemma toughness ( $r = 0.01$  in this case). Interestingly, however, one also notes that in the initial stages of the dynamics average efficiencies of cooperators grow faster than those of defectors. The reason for this is simple: As extinction pressures are larger on cooperators than on defectors, also the evolutionary pressure on inefficient cooperators is larger than on inefficient defectors (which, if favourably positioned, can occasionally generate more fitness than efficient defectors at less favourable locations). The delayed saturation of efficiencies of defectors and cooperators leads to a dynamics that is different from the usual evolution in the one-off game (cf. the filled circles in Fig.

1). Cooperators can initially gain an advantage by associating with larger efficiencies than defectors and hence they can recover from the initial decline which is caused by the assortment dynamics after starting from random initial conditions. However, the recovery of cooperation is stopped as defectors evolve towards saturation in the efficiency trait and cooperators become extinct when a homogeneous state with  $\epsilon = 1$  is reached in the entire population.

The bottom panel of Fig. 1 contrasts the co-evolution of social strategies and efficiencies for  $c = 1$  and a much more severe dilemma setting with  $r = 0.1$  to the above scenario of a free efficiency trait with  $c = 0$ . For a better visual understanding also some snapshots in the evolution which correspond to important stages of the dynamics are illustrated in Fig. 2. One first notices the difference in the asymptotic states: For costly efficiencies cooperation can survive in a regime far beyond dilemma strengths which typically support cooperation in the spatial game with  $\kappa = 0$ . Concomitantly, cooperators associate with a saturated efficiency whereas defection is typically paired with much lower values of the efficiency trait. The course of the evolution is also different from the scenario with a free efficiency trait and proceeds in several stages. First, as typical in the evolution of cooperation in spatial games, when strategies are randomly mixed cooperators are easily exploited by defectors and hence cooperation declines until assortment of like strategies is reached. In the process only cooperators with large efficiencies survive in small islands (Fig. 2 top right, which corresponds to the minimum in the number of cooperators in Fig. 1) in a sea of defectors. Within the sea of defectors low efficiency investments are favoured (as there is hardly any game payoff to be leveraged) and large efficiency defectors only survive in very small numbers when attaching to clusters of cooperators. In a second stage, large efficiency cooperators can expand into the sea of low efficiency defectors, conquering a very large share of the entire system (Fig. 2, bottom right, corresponding to the maximum in cooperation in Fig. 1). With some delay this allows large efficiency defectors to expand and eventually a stationary balance of ordered arrangements of large and low efficiency defectors and large efficiency cooperators is reached (Fig. 2 bottom left).

These initial simulations illustrate an important point: A co-evolutionary dynamics of costly efficiencies and social strategies can allow cooperation to survive far beyond the regime normally supported by network reciprocity. The origin of the support mechanism is that evolution favours efficiency enhancements in clusters of cooperators. Since cooperators benefit from surrounding themselves with like strategies, paying a cost to surround themselves with other cooperators is an evolutionary viable strategy that outcompetes the cooperate strategy that does not invest into efficiency enhancements. For defectors the situation is different. When not in contact with cooperators, defectors which invest into

efficiency enhancements are outcompeted by defectors who don't. However, only efficient defectors manage to penetrate clusters of efficient cooperators and thus a cyclic dominance (efficient cooperators beat inefficient defectors, but are beaten by efficient defectors who are in turn outcompeted by inefficient defectors) similar to Rock-papers scissors (Szolnoki and Szabo, 2004), volunteering (Szabo and Hauert, 2002) or the advertising game of (Brede, 2013a) is created. As one would expect, the balance between the three competing strategies can be shifted when modifying the cost parameter. Interestingly, however, in a large cost regime high efficiency defectors can easily be pushed into extinction and for low frequencies of recurring invasions cooperators can dominate the system over large time periods.

For a more comprehensive investigation, in Fig. 3 the phase diagrams that give the dependence of the frequency of cooperators  $n_c$  on the dilemma toughness are evaluated for low, intermediate, and high noise levels in strategy replication. Going hand in hand with this Fig. 4 gives the dependence of stationary average efficiencies on the dilemma toughness parameter  $r$ . Both, the  $n_c(r)$  curves in Fig. 3 and the  $\epsilon_C(r)$  and  $\epsilon_D(r)$  curves in Fig. 4 are given for various cost assumptions.

For the case of noiseless replication with  $\kappa = 0$  several sharp transitions can be discerned. First, comparing curves for various cost choices it is worth noting that cooperation and efficiencies can co-evolve for any cost  $c > 0$ . This is illustrated by the first panel in Fig. 3: Whereas cooperation dies out for  $r > 0$  for  $c = 0$  cooperation can survive up to around  $r \approx 0.45$  if a small cost  $c = 0.0001$  is included (and in fact in the limit  $\kappa \rightarrow 0$  in Eq. (2) any cost makes sure efficient defectors can be invaded by  $\epsilon = 0$  defectors, thus allowing for the cyclical dominance mechanism to operate). As further illustrated in Fig. 5 this is different for  $\kappa > 0$ . The more noise in strategy propagation, the larger the cost required to allow cooperation to survive. On the one hand larger costs help the evolution of cooperation since they make it easier for inefficient defectors to chase efficient defectors, hence reducing the pressure on efficient cooperators and allowing them to thrive. However, on the other hand costs above some threshold make efficiency investments unviable for both cooperators and defectors. As a consequence a range of costs exists for which cooperation is optimally supported. The dependence of cooperation on dilemma costs also includes a transition which demarcates a phase in which efficient defectors typically survive the initial stages of the dynamics from another phase in which they go extinct (cf. Fig. 5). When efficient defectors die out, the cyclical competition is replaced by a competition between efficient cooperators and inefficient defectors in which the former can dominate. Hence, for some cost range a state in which only cooperators survive is reached. This state is marked by homogeneity in agent's efficiencies, and hence it is not stable to the reinvasion of defectors. In fact, including

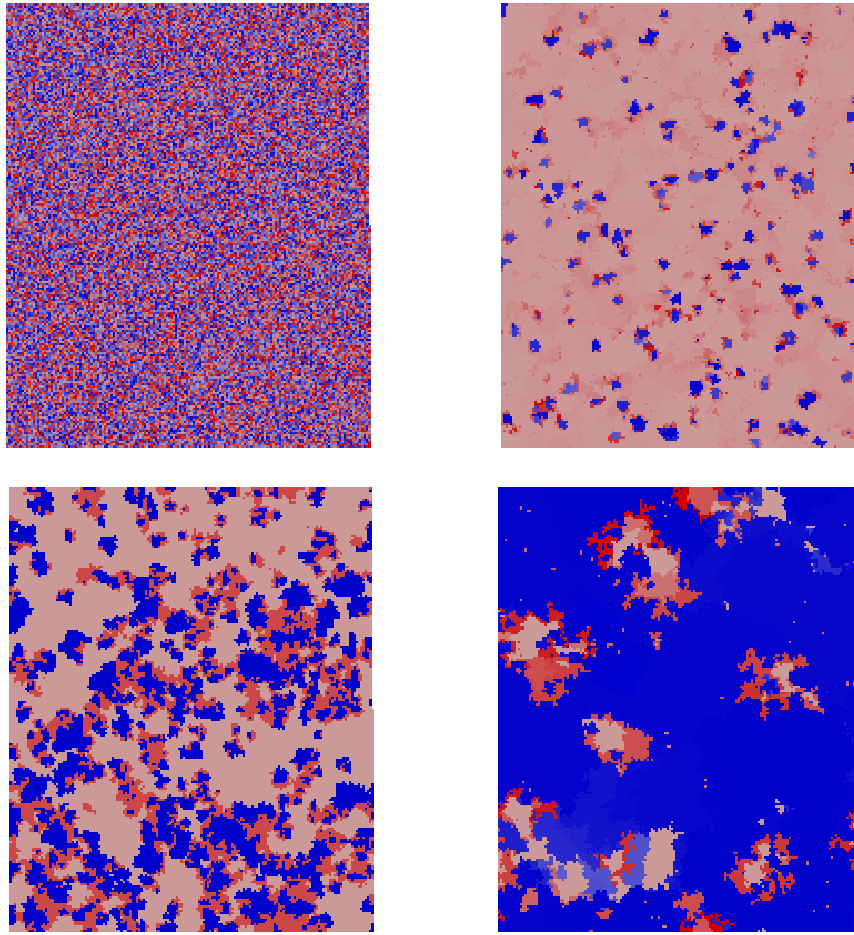


Figure 2: Typical snapshots in the arrangement of cooperators (blue) and defectors (red) at various stages of the co-evolution. Clockwise from top right to bottom left: initial conditions at  $t = 0$ , then snapshots at  $t = 42$ ,  $t = 120$ , and the asymptotic state at  $t = 3000$ . The intensity of the color of the sites indicates the efficiency trait: dark blue corresponds to cooperators with large  $\epsilon$ , light blue to cooperators with low  $\epsilon$ , and dark red and light red refer to large  $\epsilon$  and low  $\epsilon$  defectors, respectively.

invasions of agents with randomly selected strategies, large amplitude oscillations between regimes in which cooperation dominates and regimes in which invading defectors can take over large parts of the system result.

Second, for  $\kappa = 0$  the  $n_c - r$  phase diagrams in Fig. 3 and the corresponding  $\epsilon - r$  diagrams in Fig. 4 show a number of sharp transitions in the  $r$ -dependencies. Whereas cooperators always evolve into a monochromatic population (not shown), the defector population tends to become separated into groups of defectors with high ( $\epsilon = \epsilon_C$ ) and low ( $\epsilon = 0$ ) efficiency. When the dilemma strength is increased, proportions of low and high efficiency defectors shift, but the values of the group-characteristic efficiency values  $\epsilon = 0$  and  $\epsilon = \epsilon_C$  remain the same. The first order transitions in the  $n_c - r$  dependencies indicate critical values of the dilemma strengths at which sudden shifts in the relative proportions of low and high efficiency defectors take place.

Most notable, in particular for larger costs, is the tran-

sition at which high efficiency defectors become extinct (i.e. at which  $\epsilon_D \approx 0$ ). The effect is similar to what we have discussed for the  $n_c - c$  dependencies in Fig. 5 above: Without the presence of high efficiency defectors low efficiency defectors are outcompeted by high efficiency cooperators and the latter can dominate the population. Whereas efficiency investments generally decline with increasing dilemma toughness, due to the efficiency competition in the now purely cooperative population, maximum efficiencies are favoured by evolution.

With the exception of smoother transitions between the various regimes, principally similar behaviour to the case of  $\kappa = 0$  is observed for intermediate and high levels of noise. The main difference is that more noise in strategy propagation requires larger costs of the efficiency trait for cooperation to persist.

Last, it is worthwhile examining whether the co-evolutionary mechanism is robust when strategy traits are

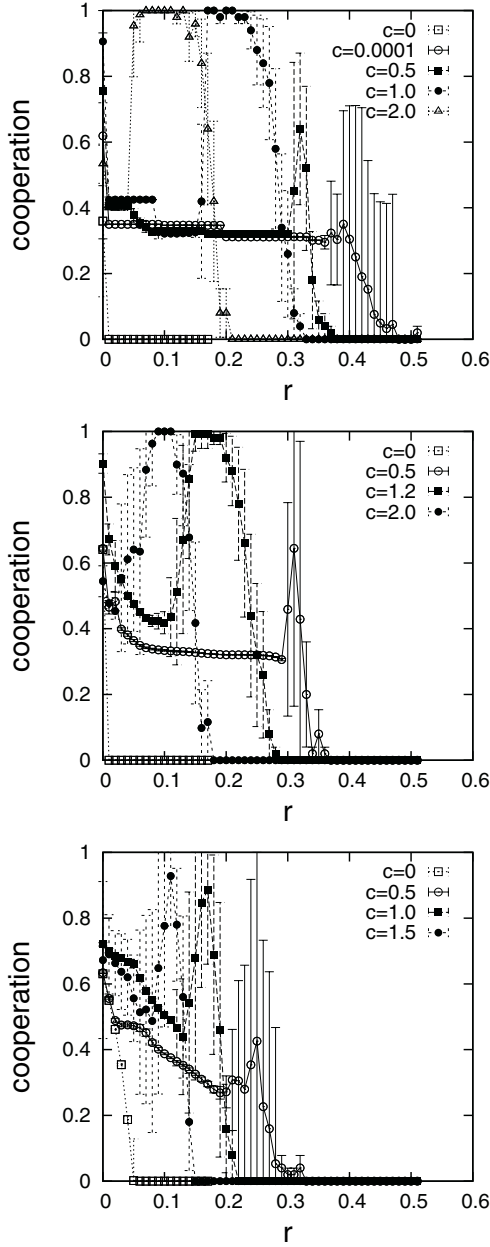


Figure 3: Dependence of the average stationary frequency of cooperators on the dilemma toughness  $r$  for noise levels in strategy updating  $\kappa = 0$  (no noise),  $\kappa = 0.1$  (low amount of noise), and  $\kappa = 1$  (large amount of noise). The dependencies are given for a range of cost parameters  $c$  for the efficiency trait. Note, that for  $c = 0$  and  $\kappa = 0$  and  $\kappa = 0.1$  cooperation can only survive for  $r = 0$  (open boxes).

inherited separately. To investigate this issue we consider an amended model in which the rules for passing on the social strategy  $s$  and the efficiency trait  $\epsilon$  are modified. If a focus agent copies from a reference agent (i.e. according to Eq.

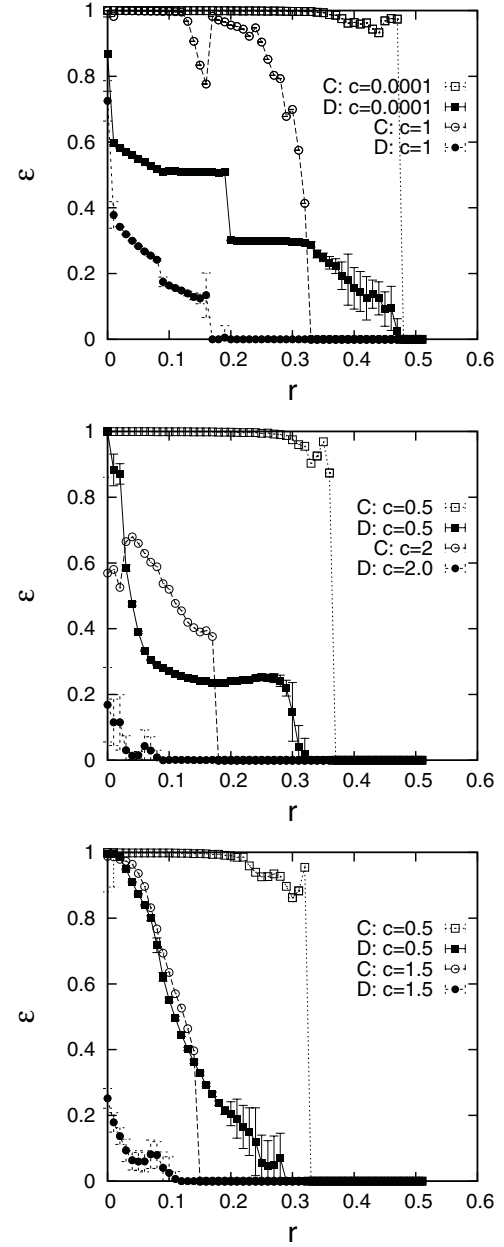


Figure 4: Dependence of the average stationary evolved efficiency traits of defectors (labelled as “D”, filled symbols) and cooperators (labelled as “C”, open symbols) on the dilemma toughness. (top) For  $\kappa = 0$  for two cost scenarios, very low cost  $c = 0.0001$  (boxes) and high cost  $c = 1$  (circles), (middle) for  $\kappa = 0.1$  and low  $c = 0.5$  (boxes) and high  $c = 2$  (circles) costs, (bottom) for  $\kappa = 1$  and low  $c = 0.5$  (boxes) and high  $c = 1.5$  (circles) costs.

(2)), with probability  $p_d$  only either the efficiency trait or the social strategy are imitated. In the opposite case, i.e. with probability  $1 - p_d$ , both traits are simultaneously passed on.



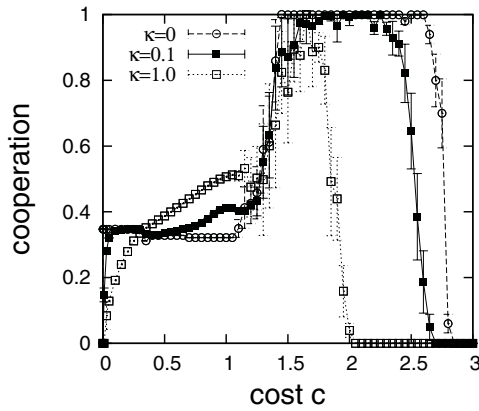


Figure 5: Dependence of cooperation on the cost of efficiencies for  $r = 0.1$  and several levels of noise in strategy propagation.

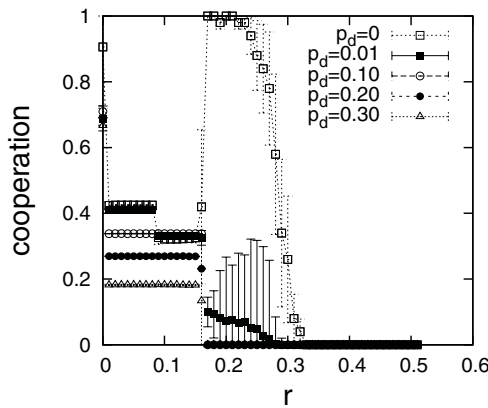


Figure 6: Dependence of cooperation on the dilemma toughness for various degrees of disjoint strategy pass  $p_d$  for  $c = 0.0001$  and  $\kappa = 0$ .

Hence the new parameter  $p_d$  classifies the degree of disjoint strategy pass, with  $p_d = 0$  corresponding to the previously considered model and  $p_d = 1$  corresponding to completely disjoint strategy pass. Figure 6 illustrates some simulation experiments in which scenarios with  $p_d > 0$  were explored. Clearly, disjoint strategy pass equalizes differences in efficiencies between cooperators and defectors, hence reducing support for cooperation. However, in contrast to the advertising game of Brede (2013a), cooperation can persist for rather substantial degrees of disjoint strategy transfer, thus adding robustness to previous results.

## Discussion and conclusions

In this paper we have considered a model for the co-evolution of social strategies and an efficiency trait that determines how well agents can convert gains from dilemma

games into evolutionary payoff. Through a series of controlled simulation experiments we have demonstrated that the co-evolution of efficiencies and social strategies can add substantial support to cooperation, if the payoff efficiency costs are within a certain range. Maximum and minimum costs that demarcate the cost window are dependent on noise in strategy replication, with lower noise generally allowing for a larger range of cooperation-supporting costs.

Even though based on a well-known cyclical dominance mechanism that has already been explored elsewhere (Szolnoki and Szabo, 2004; Szabo and Hauert, 2002), the present paper adds some significant extensions to the work of (Brede, 2013a). First, the change in the model from a trait that purely biases strategy spread to a trait that effects payoff generation adds an interesting aspect. The present paper demonstrates that pressures to enhance efficiencies are not the same for cooperators and defectors involved in evolutionary dilemmas on graphs. We demonstrate that the nature of the social game favours the evolution toward higher efficiencies in the cooperator population, and this, in turn, allows cooperation to survive.

Second, one might wonder whether the binary strategies imposed in (Brede, 2013a) constrain stationary states to settings that could not have been reached by the evolution of a continuous trait. We demonstrate here that this is not the case: A continuous efficiency trait can co-evolve with social strategies to support cooperation. The main difference compared to the binary setting is that stationary efficiency levels of the subpopulations self-organize to evolutionarily stable levels.

Third, the model presented in this paper demonstrates that the basic cooperation-supporting mechanism of (Brede, 2013a), i.e. that cooperators can afford to pay more for costly replication than defectors, is in fact more general than originally highlighted in the model based on learning and teaching. We show here that an equivalent mechanism based on costly efficiency improvements can also operate in evolutionary dynamics that are free of noise. For instance, some preliminary simulations indicate that qualitative results are robust for asynchronous updating based on ‘imitate the best’, for which no cooperation can survive in the standard spatial game (Huberman and Glance, 1993).

## Acknowledgements

The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

## References

- Abramson, G. and Kuperman, M. (2001). Social games in a social network. *Phys. Rev. E*, 63:030901.
- Beinhocker, E. (2007). *The origin of wealth*. Random House, London.

- Brede, M. (2011a). The evolution of cooperation on correlated payoff landscapes. *Artificial Life*, 17:365–373.
- Brede, M. (2011b). Playing against the fittest: A simple strategy that promotes the emergence of cooperation. *EPL*, 94:30003.
- Brede, M. (2013a). Costly advertising and the evolution of cooperation. *PLoS ONE*, available online: <http://dx.plos.org/10.1371/journal.pone.0067056>.
- Brede, M. (2013b). Short versus long term benefits and the evolution of cooperation in the prisoner’s dilemma game. *PLoS ONE*, 8(2):e56016.
- Cao, L., Ohtsuki, H., Wang, B., and Kazuyuki, A. (2011). Evolution of cooperation on adaptively weighted networks. *J. Theor. Biol.*, 272:8–15.
- Chadefaux, T. and Helbing, D. (2010). How wealth accumulation can promote cooperation. *PLoS ONE*, 5:e13471.
- Crespi, B. J. (2001). The evolution of social behavior in microorganisms. *Trends in ecology & evolution*, 16(4):178–183.
- Eshel, I. and Cavalli-Sforza, L. L. (1983). Assortment of encounters and evolution of cooperativeness. *PNAS*, 79:1331–1335.
- Hebb, D. (1949). *The organization of behaviour*. John Wiley & Sons, London.
- Huberman, B. A. and Glance, N. S. (1993). Evolutionary games and computer simulations. *Proc. Natl. Acad. Sci. USA*, 90:7716–7718.
- Masuda, N. (2007). Participation costs dismiss the advantage of heterogeneous networks in the evolution of cooperation. *Proc. R. Soc. B*, 274:1815–1821.
- Nowak, M. A. (2006). Five rules for the evolution of cooperation. *Science*, 314:1560–1563.
- Nowak, M. A. and M., M. R. (1992). Evolutionary games and spatial chaos. *Nature*, 359:826–829.
- Perc, M. and Szolnoki, A. (2008). Social diversity and promotion of cooperation in the spatial prisoner’s dilemma game. *Phys. Rev. E*, 77:011904.
- Perc, M. and Szolnoki, A. (2010). Coevolutionary games - a mini review. *BioSystems*, 99:109–125.
- Perc, M. and Wang, Z. (2010). Heterogeneous aspirations promote cooperation in the prisoner’s dilemma game. *PLoS ONE*, 5:e15117.
- Powers, S. T., Penn, A. S., and Watson, R. A. (2011). The concurrent evolution of cooperation and the population structures that support it. *Evolution*, 65:1527–1543.
- Santos, F., Pacheco, J., and Lenaerts, T. (2006a). Cooperation prevails when individuals adjust their social ties. *PLoS Comput. Biol.*, 2:1284–1291.
- Santos, F. C., Rodrigues, J., and Pacheco, J. (2006b). Graph topology plays a determinant role in the evolution of cooperation. *Proc. R. Soc. B*, 273:51–55.
- Szabó, G. and Fath, G. (2007). Evolutionary games on graphs. *Phys. Rep.*, 446:97–216.
- Szabo, G. and Hauert, C. (2002). Evolutionary prisoner’s dilemma games with voluntary participation. *Phys. Rev. E*, 66:062903.
- Szabó, G. and Toke, C. (1998). Evolutionary prisoner’s dilemma game on a square lattice. *Phys. Rev. E*, 58:69–73.
- Szolnoki, A. and Perc, M. (2008). Coevolution of teaching activity promotes cooperation. *New J. Phys.*, 10:043036.
- Szolnoki, A. and Szabo, G. (2004). Phase transitions for rock-scissors-paper game on different networks. *Phys. Rev. E*, 70:037102.
- Szolnoki, A. and Szabó, G. (2007). Cooperation enhanced by inhomogeneous activity of teaching for evolutionary prisoner’s dilemma game. *EPL*, 77:30004.
- Szolnoki, A., Vukov, J., and Szabo, G. (2009a). Selection of noise level in strategy adoption for spatial social dilemmas. *Phys. Rev. E*, 80:056112.
- Szolnoki, A., Vukov, J., and Szabó, G. (2009b). Selection of noise level in strategy adoption for spatial social dilemmas. *Phys. Rev. E*, 80:056112.
- Szolnoki, A., Wang, Z., and Perc, M. (2012). Wisdom of groups promotes cooperation in evolutionary social dilemmas. *Sci. Rep.*, 2:576.
- Szolnoki, A., Wang, Z., Wang, J., and Zhu, X. (2010). Dynamically generated cyclic dominance in spatial prisoner’s dilemma games. *Phys. Rev. E*, 82:036110.
- Tanimoto, J. and Yamauchi, A. (2010). Does “game participation cost” affect the advantage of heterogeneous networks for evolving cooperation? *Physica A*, 389:2284–2289.
- Tanimoto, J. and Brede, M. and Yamauchi, A. (2012). Network reciprocity by coexisting learning and teaching strategies. *Phys. Rev. E*, 85:032101.
- Van Segbroeck, S., Santos, F. C., Lenaerts, T., and Pacheco, J. M. (2009). Reacting differently to adverse ties promotes cooperation in social networks. *Phys. Rev. Lett.*, 102:058105.
- Van Segbroeck, S., Santos, F. C., A., N., Pacheco, J. M., and Lenaerts, T. (2008). The evolution of prompt reaction to adverse ties. *BMC Evolutionary Biology*, 8:1–8.
- Wang, Z. and Perc, M. (2010). Aspiring to the fittest and promotion of cooperation in the prisoner’s dilemma game. *Phys. Rev. E*, 82:002115.
- Wang, Z., Zhu, X., and Arenzon, J. J. (2012). Cooperation and age structure in spatial games. *Phys. Rev. E*, 85:011149.
- Weibull, J. (1996). *Evolutionary game theory*. MIT University Press, Cambridge, MA.
- Zhang, H., Small, M., Yang, H., and Wang, B. (2010). Adjusting learning motivation to promote cooperation. *Physica A*, 389:4734–4739.
- Zimmermann, M. and Eguíluz, V. M. (2005). Cooperation, social networks, and the emergence of leadership in a prisoner’s dilemma game with adaptive local interactions. *Phys. Rev. E*, 72:056118.

# Multiple Life-History Stage Competition and its Effect on Coexistence.

Miguel Gonzalez<sup>1</sup>, Richard Watson<sup>1</sup>, Jason Noble<sup>1</sup> and Patrick Doncaster<sup>2</sup>

<sup>1</sup>Institute for Complex Systems Simulations, University of Southampton, UK

<sup>2</sup>Faculty of Natural and Environmental Sciences, University of Southampton, UK  
mgc1g11@soton.ac.uk

## Abstract

Explaining the long-term coexistence of many species in a complex ecosystem has been an important topic in both A-Life and ecology for several decades. Neutral and niche theories have been developed in parallel to explain ecological patterns of coexistence. Among the niche theories, trade-offs between species seem to play important roles in the mechanism of most models proposed so far. One of the many trade-off approaches to explain coexistence within trophic levels explores the scenario of species having two developmental stages existing in different ecological niches. Previous work has shown that such multi-stage models can sustain many species with inter-specific competitive coefficients larger than intra-specific ones for one of the stages, but the effect of such scenarios on the possible combinations for coexistence in the parameter space has not been explored. Here, we build on previous work by considering the effect of adding more stages to the competition model and analysing the relative sizes of the basins of attraction leading to coexistence. Computational simulations and Monte Carlo methods were used to analyse each number-of-life-stages case. The results show an increase in the number of coexistence cases between one and three stages. For more than three stages coexistence cases are reduced in relation to the parameter space due to the averaging effect of multiple competitive coefficients. The implications of such results could offer a potential explanation for coexistence patterns in ecology and adaptive ones in evolutionary biology.

## Introduction

The successful modeling of open-ended complexity in real ecosystems has long been a target for A-Life researchers (Channon and Damper, 2000; Channon, 2008; Shao and Ray, 2010; Ray, 1994). One of the great challenges for these models is achieving the long-term coexistence of a diverse collection of species. All too often the model collapses into a mono-culture or a small trophic cycle. One of the many simplifications that such models typically employ is to consider only a single life-stage for each of the species involved: there are only tadpoles, for example, and not both tadpoles and frogs. Here we show that including the possibility of multiple life stages can actually *promote* the persistent coexistence of multiple species, as Moll and Brown (2008)

have previously argued. Furthermore, we show that there is an optimal number of life stages to achieve long-term complexity, and that this number is a function of how quickly the environment fluctuates.

In current ecological literature a great number of theories and models have been developed in order to explore different biodiversity patterns (Chesson, 2000). Species coexistence models are concerned primarily with the basic problem of resource allocation among different species and the principle of competitive exclusion. These models attempt to generate *stable* coexistence over extended periods of time and they contrast with models of *unstable* coexistence which are concerned with mechanisms for delaying competitive exclusion by minimizing the differences between species fitness (Chesson, 2000; Moll and Brown, 2008). All *stable* coexistence models operate by establishing directly or indirectly a greater degree of intra-specific competition compared to inter-specific competition (Chesson, 2000).

Multiple Life-History Stage models are a type of stable coexistence model that depend on trade-offs. Here, the trade-off happens between different developmental stages of two or more species. In particular, these trade-off models show that the apparent coexistence of a great number of species is explained by the presence of many life-history stages with associated ecological niches. This is a phenomenon that has not yet been considered in the A-Life literature, and so in the current paper we draw on the relevant ecological literature to develop a simple dynamical-systems model of competing species with multiple life stages.

During their development, organisms of several species go through a series of phenotypic changes that affect the way they interact and exploit their environment (Werner and Gilliam, 1984). Drastic examples of these cases can be seen in insects (Dopman et al., 2007), amphibians (Werner and McPeck, 1994; Werner et al., 1995; Werner and Anholt, 1996) and fish (Arendt and Wilson, 1997). But similar interpretations can be extrapolated to plants with their different seed, sprout, juvenile, and adult stages, and plankton where different fluid dynamics and predatory pressures affect their interaction with the medium (Padisák et al., 2003). From

an evolutionary point of view, these differences between developmental stages allow juveniles to avoid direct competition with adults of the same species, and following Gillian's rule, evolution can move the population towards niche shifts where the ratio of mortality over individual growth is minimized (Werner and Gilliam, 1984; Claessen and Dieckmann, 2002). This process of adaptive niche shift during individual development is known as ontogenetic niche shift. From a competitive point of view, the picture that such systems show is one with an age-structured population where different species in the same guild and with similar developmental stages go through a stepped competition sequence where some individuals might do better than others at different stages (Moll and Brown, 2008; Fujiwara et al., 2011). The second diagram in figure 1 explains this approach for the simplest case of two species and two stages.

Previous work on this approach includes Moll and Brown (2008) and Fujiwara et al. (2011). But analysis on the coexistence space that such systems generate for every possible combination of competitive effects has not been explored. As a result, the hypothesis of coexistence in the parameter space being increased by ontogenetic niche shift has not been tested. Moll and Brown (2008) has proposed a very simple model to explain how coexistence could be possible in cases where inter-species competitive coefficients are greater than intra-specific ones (provided that this situation was restrained to only one life stage). An even more detailed model of the two-life-stages scenario is explored by Fujiwara et al. (2011), but in none of these cases has the extension to more life stages and the effect of ontogenetic niche shift on coexistence space been explored. In order to answer these questions, we propose a simplified version of Moll's model and we estimated the proportion of the hyper-dimensional parameter-space in which coexistence happens by using Monte Carlo techniques (Kroese et al., 2011) for a range of life-stage numbers.

### The Model

The most basic version of the model is described by two competing species with two life stages each, with competition between the life stages of a single species assumed to be absent. Equally, competition between the life stages of two different species is also assumed to be absent. Similar cases can sensibly be assumed to happen in nature in several species (Werner and Hall, 1988; Werner and McPeck, 1994; Werner and Anholt, 1996; Arendt and Wilson, 1997). The lack of competition between species and stages is an assumption for the sake of keeping the model tractable. Such competition could of course exist in the real world, but is not relevant to the question being asked here. Figure 1 shows the basic diagram for a single-stage, two-stage and three-stage competition models.

The discrete time-step version of the system shown in Figure 1 can be expressed with the following system of equations

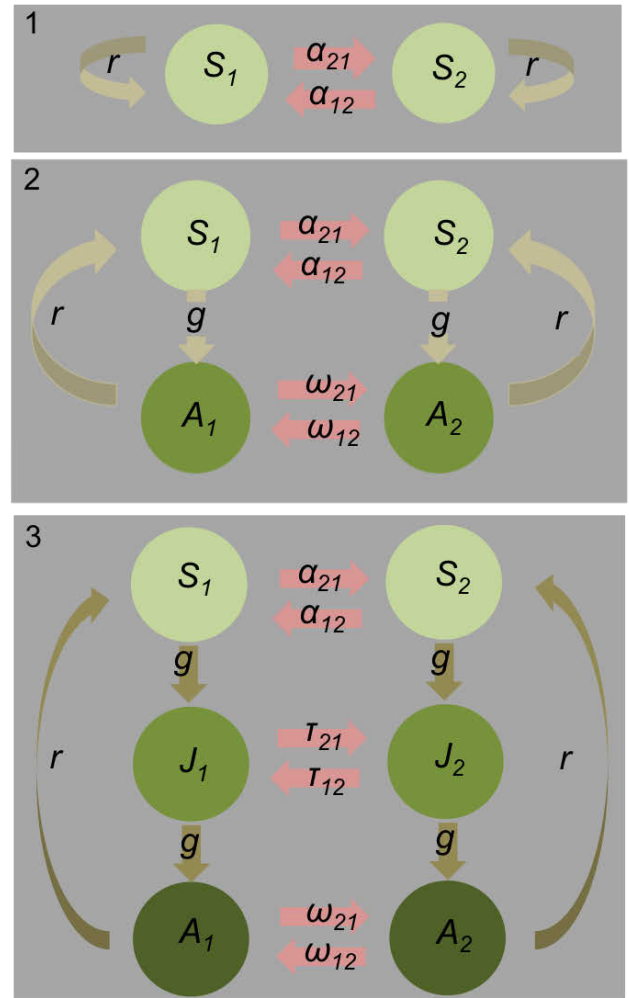


Figure 1: Diagrams for 1-stage, 2-stage and 3-stage models. Circles  $S$  and  $A$  represent the first and last stages respectively,  $J$  represents an intermediate stage between  $S$  and  $A$ ,  $r$  is the adult reproductive rate and  $g$  is the individual growth rate, and competition coefficients are represented by red arrows between circles, ( $\alpha$ ,  $\omega$  and  $\tau$ ). The model includes competitive effects in both directions and its values are standardized in relation to intra-specific competition. For instance,  $\alpha_{ij}$  corresponds to the effect of species  $j$  on species  $i$  in the first stage of development.

tions for an objective species  $i$ . These are modified and extended versions of Moll's model (Moll and Brown, 2008) which is itself based on a two species Ricker (1954) model.

$$S_{i,t+1} = r_i A_{i,t} e^{-(A_{i,t} + \omega_{ij} A_{j,t})} + (1-g) S_{i,t} e^{-(S_{i,t} + \alpha_{ij} S_{j,t})} \quad (1)$$



$$J_{i,t+1} = g(p_{i,t})e^{-(p_{i,t})+\chi_{ij}(p_{j,t}))} + (1-g)J_{i,t}e^{-(J_{i,t}+\tau_{ij}J_{j,t})} \quad (2)$$

$$A_{i,t+1} = g(p_{i,t})e^{-(p_{i,t})+\chi_{ij}(p_{j,t}))} + A_{i,t}e^{-(A_{i,t}+\omega_{ij}A_{j,t})} \quad (3)$$

Where  $S_{i,t}$  is the density of seeds of species  $i$  at time-step  $t$ ,  $A_{i,t}$  is the density of adults of species  $i$  and  $J_{i,t}$  is the corresponding to middle stages;  $r_i$  is the rate of growth from adults to seeds,  $g_i$  is the growth rate between stages.  $\alpha_{ij}$  is the competition effect of seeds of species  $j$  on species  $i$ ,  $\omega_{ij}$  is the equivalent for adults. The equation in the middle (Eq. 2) is the generic expression for any stage different from first or last; the value  $p_{i,t}$  corresponds to the density of individuals in the previous stage for time  $t$ . Concordantly, the parameter  $\chi_{ij}$  is a generic variable that corresponds to the competition coefficients in the previous stage. Intuitively it follows that  $\tau_{ij}$  refers to the competitive coefficient of stage  $J$ . Extensions to more stages follow by incorporating new density parameters and competition coefficients. The effects on every life stage are calculated accordingly considering the proportion of individuals that stay at that stage and the proportion that move to the next one. These systems of equations are deterministic but analytic solutions can not be found (Hassell and Comins, 1976). It is possible to look for isoclines in the tetra-dimensional space of the 2-stage model, but since the interest is in investigating percentages of parameter space that converge on coexistence for many different number of stages, a simulation approach was selected. In order to explore this system some simplifications were done in relation to the original Moll and Brown model. The effects of differences in growth rate and reproduction rate between species are not considered. For this reason, reproduction rates and growth rates are fixed as the same for both species, in this case defined by  $g$  and  $r$ . The individual growth rate between stages within a species is also considered as the same: in other words, individuals develop at a constant speed. The model also assumes the intra-specific competition to be 1.0 and the top density for any stage (determined by the load capacity of the system) is also assumed to be 1.0 (see Moll and Brown, 2008, for more information). Scaling terms are not considered following the assumption that competitive coefficients are relative to intra-specific ones.

## Methods

### Model Dynamics

In order to follow the effect of competition coefficients on the dynamics of the model the values of  $r$  and  $g$  were fixed,

on  $r = 1.5$  and  $g = 0.5$  (unless specified differently). The argument for the selection of these values was based on a preliminary exploration and will be discussed later. The simulations were run for 150,000 iterations or until the system reached equilibrium. This number of iterations was selected after running simulations across the parameter space and noticing no important differences between increases in iterations from this value on. The result of a particular simulation was assumed to be coexistence if no species became extinct (density under 0.00001 on at least one stage). In cases where one of the species turned out to be excluded the identity of the species was recorded. Initial densities or quantities were selected randomly for all simulations. To explore the most basic 2-stage version of the model, four conditions were selected for the  $\alpha_{ij}$  and  $\alpha_{ji}$  values. In the first condition both alphas were set to 1.0, making the intra-specific competition the same as the inter-specific one for the first stage. In the second case alpha values were selected below 1.0, making the first stage a coexistence scenario. The third case sets one alpha as greater than 1.0 and the other as less than 1.0, making the initial stage a competitive exclusion scenario. Finally, the last case considers both alpha values as larger than 1.0, which in a single-stage scenario will create alternate states dependent on initial density conditions for the four quantities. On each of these conditions a full 2D landscape for  $\omega_{ij}$  and  $\omega_{ji}$  was explored. The range explored was between omega values of 0.0 and 2.0 with a resolution of 0.01; at each of these points 50 simulations were run starting from random initial quantities for  $S_i$  and  $A_i$ , creating 200 X 200 x 50 simulation plots as shown in Figure 2 in the results section. These results can be considered a replication of those obtained by Moll and Brown (2008). As shown by Wilbur (1996) and Moll and Brown (2008) the model exhibits a range of behaviours from stable equilibrium to chaos to oscillations. In this work the range of parameters selected does not show any other behavior except simple attractors. In this case the space of exploration for the competitive coefficients goes between 0.0 and 2.0. Individual growth rates  $g$  between 0.01 and 0.99 were explored, as well as adult intrinsic reproduction rates  $r$  between 0.95 and 2.5.

### Exploring the 2-Stage Model

Each of the plots shown in Figure 2 could be interpreted as a slice of a tetra-dimensional space, where each dimension is characterized by the competition coefficients. On these slices the area of coexistence is a continuum square area that starts in the origin of the space (Moll and Brown, 2008, and Figure 2). With this knowledge, an estimation of the total hyper-volume of coexistence in the parameter space (from 0.0 to 2.0 on each dimension) could be obtained by estimating the omega values for which equilibrium coexistence stops being an attractor on both omega coefficients. In order to do this, one omega value is kept at 0.0 while the other explores simulations with values from 0.0 to 2.0 until coexis-



tence disappears. Once this has been done with both omega coefficients the proportion of the total coexistence area for that slice can be calculated by multiplying both omega values where coexistence does not happen any more, and then dividing this area over the  $2.0 \times 2.0$  area of search. Doing this with every combination of alpha values creates a 3D landscape where each combination of alphas have an associated percentage of coexistence space in the omega slice. The total space can then be estimated by averaging all the percentages and in this way we can calculate an estimate that can be checked with the Monte Carlo simulation results that will be explained in the next section. This 3D landscape is shown in Figure 3.

### Monte Carlo Simulations

In order to explore systems with more stages, a sensible way forward is taking random samples in the hyper-dimensional space and seeing in what proportions of these simulations coexistence is obtained. This approach is known as Monte Carlo search or ‘probing’. To do this, a large enough number of points should be selected to make a representative estimation. A set of tests were run using different numbers of points, and replicates. The conclusion was that 10,000 points and 10 replicates seemed to offer a good balance between small standard deviations and short simulation times. For each number of life stages between 1 and 10, Monte Carlo searches were performed (Figure 4). Also the effect of  $g$  and  $r$  values was tested and no qualitative differences were found.

### Results

Results are divided in two parts: first an exhaustive exploration of the basic 2-stage 2-species model is explored (Figure 2 and Figure 3). Secondly the results for the Monte Carlo simulations will show the coexistence percentage of the parameter space for systems with different numbers of stages (Figure 4).

#### Coexistence in the 2-Stage 2-Species Model

The simple 2-stage 2-species model was explored for 4 different scenarios regarding the competitive coefficients of the first stage; a similar approach was followed in Moll’s work. In the first case (Figure 2.A) the competitive coefficients for the first stage, or alpha values, were set to  $\alpha_{12} = 1.0$  and  $\alpha_{21} = 1.0$ . In this scenario the intra-specific and the inter-specific competitive coefficients for the first stage are exactly the same, which makes the system analogous to the 1-stage case, creating a 25% surface of coexistence for the omega slice. The second scenario (Figure 2.B) corresponds to the case where  $\alpha_{12} = 0.75$  and  $\alpha_{21} = 0.5$ ; these values were selected arbitrarily keeping in mind that they should be lower than one. Under this scenario the first stage suggests a coexistence case where inter-specific competition is lower than

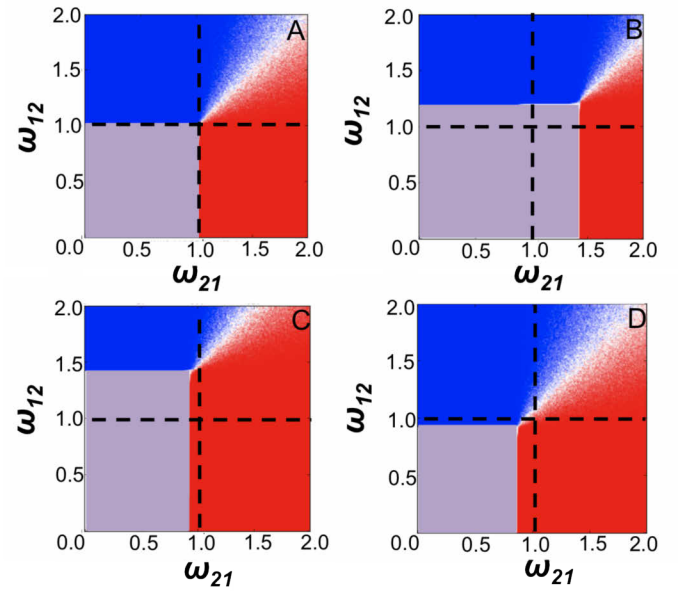


Figure 2: Results for combinations of  $\omega$  values considering four different scenarios for  $\alpha$  values. Values were explored from 0.0 to 2.0 with a step resolution of 0.01, creating  $200 \times 200$  plots. Each point in this plot is the outcome of 50 simulations starting on random initial conditions for quantities  $S_i$  and  $A_i$ . For each simulation that species 1 won the point goes towards a red shift, for each simulation species 2 won the same happens towards blue. Coexistence is represented by purple, and in every instance is an attractor point (50 replicates converge to it). The figure shows how the coexistence region for the  $\omega$  space can be expanded depending on the values of  $\alpha$ . Plot A shows a case where competitive coefficients are equal to intra-specific ones, rendering the first stage ineffective in terms of outcome  $\alpha_{12} = 1.0$ ,  $\alpha_{21} = 1.0$ . The B plot shows a case where both alpha values suggest coexistence,  $\alpha_{12} = 0.75$ ,  $\alpha_{21} = 0.5$ . C shows the case where competitive exclusion would happen considering only the first stage  $\alpha_{12} = 0.5$ ,  $\alpha_{21} = 1.15$  and D shows the case where the initial stage would suggest an alternate state scenario  $\alpha_{12} = 1.25$ ,  $\alpha_{21} = 1.15$ .  $g = 0.5$ ,  $r = 1.5$ . Similar results can be found in Moll and Brown (2008).

the intra-specific. Such a case increases the area of coexistence when compared to the first scenario. Figure 2.C shows the omega space for the case  $\alpha_{12} = 0.5$  and  $\alpha_{21} = 1.15$ ; in this scenario the first stage suggests a competitive exclusion by species 1. In the space of possible second stage competitive coefficients this translates into a contraction of the coexistence space in the  $\omega_{21}$  axis and an expansion in the  $\omega_{12}$  axis, in relation to the first scenario. Finally, Figure 2.D shows the case where alpha values suggest alternate states dependent on initial conditions for the quantities  $S_i$  and  $A_i$  for every species  $i$ ; in this case the alpha values correspond

to  $\alpha_{12} = 1.25$  and  $\alpha = 1.15$  which creates a contraction in the coexistence area in relation to the first scenario.

Considering that the coexistence space remains as a continuous hyper-volume centred on the origin, a more exhaustive exploration of the space was performed. A plot that shows what percentage of the omega space (shown in the four scenarios of Figure 2) is composed by the coexistence region, per each combination of alpha values, was generated. Figure 3 shows such a plot, where the nonlinear nature of the coexistence space expansion for lower values of alpha justifies the increase in coexistence space in the 2-stage system compared to the 1-stage version. The implications of this, and its relation to further results in Figure 4, will be discussed below.

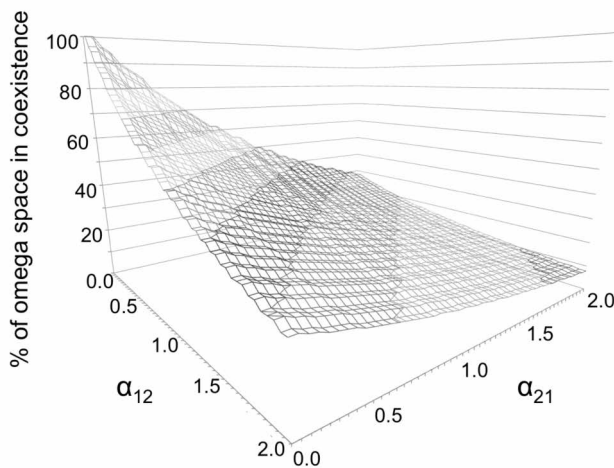


Figure 3: The figure shows the alpha space, where every combination of alpha has an estimated percentage of coexistence area in the omega space (Figure 2). The figure shows the non-linearity in the growth of the coexistence space for low alpha values. This in turn shows that the magnitude in cases where the coexistence area expands in relation to the 1-stage model (25% coexistence) is greater than the contraction in other cases.  $g = 0.5$   $r = 1.5$

### More Than Two Stages

Monte Carlo simulations for a range of multiple stage systems (from 1 stage to 10 stages) were run and the results are shown in figure 4; each bar represents the average value of 10 Monte Carlo runs with 10,000 random points in the entire parameter space; initial conditions for density quantities are randomly selected. Error bars, both positive and negative, represent a single standard deviation from the average. The decrease in coexistence space after 3 stages is due to the averaging effect of systems with many stages and it will be

discussed and explained in the next section. Dimensionality of the parameter space increases with stages but the number of points considered in the Monte Carlo simulations is enough for the standard deviations not to grow beyond sensible ranges.

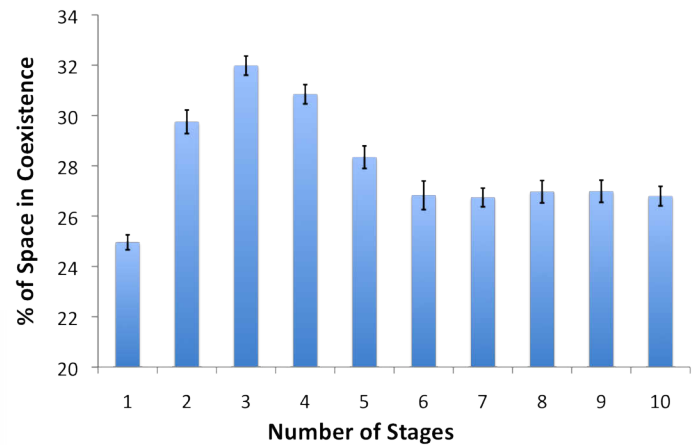


Figure 4: Percentages of the hyper-dimensional parameter space for competition coefficients that converges to coexistence. Results obtained by 10 replicates of 10,000 point Monte Carlo searches. The error bars correspond to a single standard deviation. The proportion of the parameter space that converges to coexistence seems to be increased around a 3-stage scenario and then drop down when more stages are added.  $g = 0.5$ ,  $r = 1.5$ .

## Discussion

Ontogenetic niche shift is a well-documented process that produces a niche separation, and in some cases isolation, of different developmental stages in an organism's life. Previous theoretical work has shown that such processes can be interpreted as an adaptation that arises from the reduced intra-specific competition and associated fitness gain for individuals that exhibit some degree of separation between life-stage niches (Claessen and Dieckmann, 2002). In nature, cases of ontogenetic niche differences are abundant in plants, insects, amphibians, and fish. These cases offer a very clear picture of what it means to have two separate life stages with different niches. For instance, adult frogs and tadpoles live in different ecological niches, predated different sources of food, and being predated by different trophic levels. Similar cases can be seen in numerous species of fish that, during their development, move to different trophic levels and as a result to different ecological niches within the ecosystem (Werner and Hall, 1988). In the case of plants, which are of particular interest in coexistence research, different life stages can also be present. For instance, the environmental constraints and challenges during germination and plantule stages could be very different to the ones

present as fully grown adult plants (Eriksson, 2002). Although these scenarios strike us as obvious cases of ontogenetic niche shift, it seems hard to argue that coexistence of a large number of species in big communities (such as tropical forests or plankton communities) happens due to the effect of this characteristic on every single species evaluated. With respect to this point it is important to point out that such stages might not need to be obvious in the morphological sense for them to be functionally present. Certain ecological dynamics in terms of prey size (Mittelbach et al., 1988) and locomotion ability (Padisák et al., 2003) can happen with very small variations in phenotypic characters. In the same way, light capture and soil dependence could shift suddenly between different plant niches (Auffret et al., 2010). In other words, the presence of morphologically obvious differences between stages is an indication, but not a condition, for having different ecologically functional stages during development. Because the values for competition coefficients are relative to the intra-specific competition for that stage, and similarly individual growth and adult per capita reproductive rates are also relative to these values, similar results can be obtained with different competition coefficients and values for  $g$  and  $r$ .

### The 3 Stages Bump Explained

The most interesting result shown here is the increase in the size of the coexistence region that exists in the parameter space systems with 2, 3 and 4 stages. After this increase the coexistence region seems to shrink again when more stages are added (Figure 4). In this pattern there seems to be two different effects at work. First, the addition of more stages seems to increase the proportion of space that corresponds to coexistence; this is shown by the results in Figure 3, where the nonlinear increase of coexistence space for low  $\alpha$  values is bigger than the contraction for bigger ones, as in the case shown in Figure 2.D. Following this example it would be intuitive to assume that more stages allow for more coexistence; in a biological sense this translates to more scenarios where good outcomes in certain stages can maintain a species' survival despite bad outcomes in other stages. The drop in coexistence when adding more than three stages comes about because of a counteracting trend that happens when selecting random values for the competition coefficients. In order to understand the problem in an intuitive way, consider that every multi-stage system can be equivalent to a 2-stage case with different growth and reproduction rates. In Figure 5 a diagram explaining how to build an analogous 2-stage model from a multi-stage one shows how the randomly selected competition coefficients for the first stages tend to average around 1.0 for sampling between 0.0 and 2.0. This creates an analogous model where the first-stage competition is neutral to the outcome of the dynamics. In this way the incorporation of more stages approximates the model to the one described in Figure 2.A, where the

coexistence volume is close to 25%. This result is not an artifact in the sense that the Monte Carlo simulation is not biased in any way; the parameter space “eats” itself when more dimensions are added.

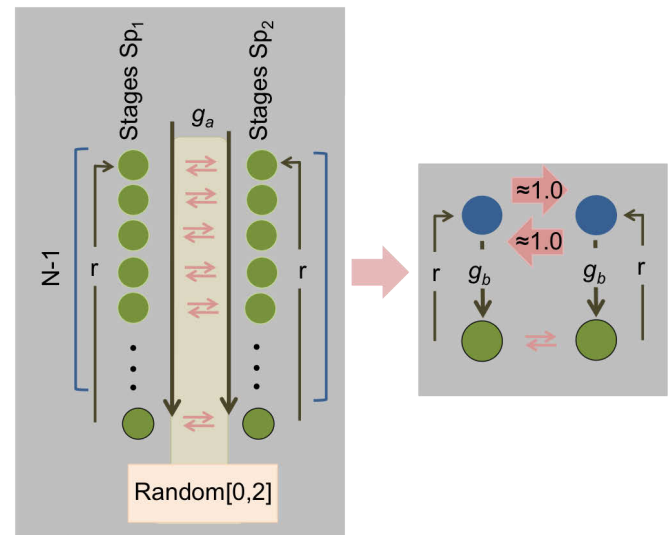


Figure 5: The detrimental effect of multi-stage systems on coexistence considering a system with  $N$  stages (left side); the random search for competitive coefficients in the space  $[0,2]$  will converge to a scenario where the inter-specific competition for the first  $N-1$  stages are analogous to a single stage with values close to 1. As shown in Figure 2.A this case represents a similar proportion to the single stage case, with only 25% of the space converging to coexistence. The only difference would be that in the analogous case the individual growth rate will be reduced due to the compression caused by merging  $N-1$  stages ( $g_a > g_b$ ), but this would not have an impact on single stage models.

Another way to understand this outcome is by considering the hypothetical case where the number of stages is infinite; in this case we would be looking at a continuous life development where competition only happens between same-age individuals. That scenario could also be analogous to noisy competitive abilities during development. In any of these two interpretations the effects of the two-stage niche segregation are lost, and the competitive coefficients can be averaged and considered as a single stage model, with 25% coexistence space in the range  $[0,2]$  for competitive coefficients. In nature, it can be sensibly assumed that over long periods of time the competition coefficient from the point of view of an objective species in relation to every competitor in a particular niche can vary. This would thereby create a pseudo-random sampling of competitive coefficients for every stage, over evolutionary timescales. It is tempting to state that under this consideration, the apparent presence of no-more-than-3 stages in most species with obvious ontoge-

netic niche shifts could be assumed as an adaptation for individuals of species that experience long term fluctuations of competitors in their different developmental niches. In order to test this hypothesis further work needs to be done, specifically taking into consideration the effects of reduction of intra-specific competition and co-evolutionary dynamics in multiple-species scenarios. Nevertheless, the results shown here suggest that such an idea could have the potential to explain such a trend in nature.

### Competition Between Stages

After a quick review of the results it might be apparent that this model relies heavily on the niche isolation between life-stages, which translates into diminished or absent vertical competition between stages. The effects of the inclusion of vertical competition will affect the results observed here depending on the magnitude considered. Hypothetically an equivalent scenario can be obtained if  $g$  and  $r$  rates are increased in certain stage transitions relative to others. The incorporation of vertical competition coefficients could show potential to explore a continuous spectrum with different degrees of isolation between stages. The expected result of such a model would be that at one end of the spectrum, where the impact of vertical (and diagonal) competition is fully considered, the model will converge back to the 1-stage scenario considering the competitive coefficient values as a function of all the coefficients involved. At the other end of the spectrum the effect uncovered by our model would be observed.

### More Than Two Species

In this work the considerations regarding competitive intransitivity between species were not analyzed. Nevertheless, the model proposed can be considered, in a simplified scenario, as a multiple-species case for any objective species  $i$ . Under this interpretation the effect of the second species can be said to be the weighted effects of all the species present in a particular niche. This is just an interpretation of the model, but in terms of the dynamics it does not capture the full effect of the competitive relations between species. Further work should be done in order to accommodate these cases.

### Natural Selection as a Promoter of Biodiversity

As a final point of discussion, it is important to consider in parallel the implications of: a) the results obtained in this work and b) the nature of ontogenetic niche shifting as an adaptation. As shown by Claessen and Dieckmann (2002) ontogenetic niche shifting can be seen as an individual adaptation responding to a pressure to reduced intra-specific competition. At the same time, ontogenetic niche shifting can potentially be seen as an adaptation to reduced inter-specific competition for objective species in changing environments. Such conclusions lead to a scenario where a process driven by natural selection at one scale promotes

coexistence and by extension biodiversity at another scale. Usually, natural selection has been seen as a process that reduces diversity in classic evolutionary theory. A further analysis of processes like the ones discussed here could show that under certain conditions natural selection can actually provide the basis for biodiversity rather than constrain it. Further analysis and model development to test this idea should be considered, but the result shown here seems to indicate that this could be the case.

### Conclusions

A review and classification of the current competition models that explain species coexistence framed the multiple life-history-stage version as a trade-off model. It was shown that such a model increases the percentage of the parameter space that converges to coexistence equilibrium points in cases with two, three and four stages. The effect is diminished by the incorporation of more than four stages due to the averaging effect of randomly occurring competitive coefficients, which effectively reduce the model to a single-stage case. The innovation of this piece of work relies on the approximation used to evaluate the coexistence space and the discovery of a non-linear pattern between the proportion of coexistence space and the number of stages. A set of more evolutionarily oriented simulations should be explored to determine the soundness of an adaptive explanation for biological species having no more than a particular number of life stages.

### Acknowledgements

This work was supported by an EPSRC Doctoral Training Centre grant EP/G03690X/1.

### References

- Arendt, J. D. and Wilson, D. S. (1997). Optimistic growth: Competition and an ontogenetic niche-shift select for rapid growth in pumpkinseed sunfish (*leporomis gibbosus*). *Evolution*, 51(6):1946–1954.
- Auffret, A. G., Meinen, E., Bruun, H. H., Ejmaes, R., and Graae, B. j. (2010). Ontogenetic niche shifts in three *vaccinium* species on a sub-alpine mountain side. *Plant Ecology and Diversity*, 3(2):131–139.
- Channon, A. (2008). A measure for natural selection's contribution to the origins and maintenance of organismal complexity. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*.
- Channon, A. D. and Damper, R. I. (2000). Towards the evolutionary emergence of increasingly complex advantageous behaviours. *International Journal of Systems Science, special issue on Emergent Properties of Complex Systems*, 31(7):843–860.
- Chesson, P. (2000). Mechanisms of maintenance of species diversity. *Annual Review of Ecology and Systematics*, 31:343–366.



- Claessen, D. and Dieckmann, U. (2002). Ontogenetic niche shifts and evolutionary branching in size-structured populations. *Evolutionary Ecology Research*, 4:189–217.
- Dopman, E. B., Sword, G. A., and Hillis, D. M. (2007). The importance of the ontogenetic niche in resource-associated divergence: Evidence from a generalist grasshopper. *Evolution*, 56(4):731–740.
- Eriksson, O. (2002). Ontogenetic niche shifts and their implications for recruitment in three clonal vaccinium shrubs: *Vaccinium myrtillus*, *vaccinium vitis-idaea*, and *vaccinium oxycoccos*. *Canadian Journal of Botany*, 80(6):635–641.
- Fujiwara, M., Pfeiffer, G., Boggess, M., Day, S., and Walton, J. (2011). Coexistence of competing stage-structured populations. *Nature, Scientific Reports*, 1(107).
- Hassell, M. P. and Comins, H. N. (1976). Discrete time models for two-species competition. *Theoretical Population Biology*, 9:202–221.
- Kroese, D., Taimre, T., and Botev, Z. (2011). *Handbook of Monte Carlo Methods*. New York: John Wiley and Sons.
- Mittelbach, G. G., Osenberg, C. W., and Leibold, M. A. (1988). Trophic relations and ontogenetic niche shifts in aquatic ecosystems. In Ebenman, B. and Persson, L., editors, *Size-Structured Populations*. Springer-Verlag Berlin Heidelberg.
- Moll, J. D. and Brown, J. S. (2008). Competition and coexistence with multiple life-history stages. *American Society of Naturalist*, 171(6):839–843.
- Padisák, J., Soróczki-Pintér, É., and Rezner, Z. (2003). Sinking properties of some phytoplankton shapes and the relation of form resistance to morphological diversity of plankton — an experimental study. *Developments in Hydrobiology*, 171:243–257.
- Ray, T. S. (1994). An evolutionary approach to synthetic biology: Zen and the art of creating life. *Artificial Life*, 1(1/2):179–209.
- Ricker, W. (1954). Stock and recruitment. *Journal of the Fisheries Research Board of Canada*, 11:559–623.
- Shao, J. and Ray, T. S. (2010). Maintenance of species diversity by predation in the tierra system. In Fellermann, H., Hanczyc, M. D. M. M., Laursen, L. L., Maurer, S., Merkle, D., Monnard, P., and Rasmussen, K. S. S., editors, *Artificial Life XII: Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pages 533–540. MIT Press, Cambridge, MA.
- Werner, E. and Anholt, B. R. (1996). Predator-induced behavioral indirect effects: consequences to competitive interactions in anuran larvae. *Ecology*, 77:157–169.
- Werner, E. and McPeck, M. (1994). Direct and indirect effects of predators on two anuran species along an environmental gradient. *Ecology*, 75:1368–1382.
- Werner, E. E. and Gilliam, J. F. (1984). The ontogenetic niche and species interactions in size-structured populations. *Annual Review of Ecology and Systematics*, 15:393–425.
- Werner, E. E. and Hall, D. J. (1988). Ontogenetic habitat shifts in bluegill: The foraging rate-predation risk trade-off. *Ecology*, 69:1352–1366.
- Werner, E. E., Wellborn, G. A., and McPeck, M. A. (1995). Diet composition in postmetamorphic bullfrogs and green frogs: implications for interspecific predation and competition. *Journal of Herpetology*, 29:600–607.
- Wilbur, H. M. (1996). Multistage life cycles. In Rhodes, O. J., Chesser, R. K., and Smith, M. H., editors, *Population Dynamics in Ecological Space and Timepace and time*. University of Chicago Press.



# Robustness and Directed Structures in Ecological Flow Networks

Taichi Haruna<sup>1</sup>

<sup>1</sup>Department of Earth & Planetary Sciences, Graduate School of Science, Kobe University,  
1-1, Rokkodaicho, Nada, Kobe 657-8501, Japan  
tharuna@penguin.kobe-u.ac.jp

## Abstract

Robustness of ecological flow networks under random failure of arcs is considered with respect to two different functionalities: coherence and circulation. In our previous work, we showed that each functionality is associated with a natural path notion: lateral path for the former and directed path for the latter. Robustness of a network is measured in terms of the size of the giant laterally connected arc component and that of the giant strongly connected arc component, respectively. We study how realistic structures of ecological flow networks affect the robustness with respect to each functionality. To quantify the impact of realistic network structures, two null models are considered for a given real ecological flow network: one is random networks with the same degree distribution and the other is those with the same average degree. Robustness of the null models is calculated by theoretically solving the size of giant components for the configuration model. We show that realistic network structures have positive effect on robustness for coherence, whereas they have negative effect on robustness for circulation.

## Introduction

Networks have been usually considered as undirected in the field of complex networks (Newman, 2003). However, many real-world networks are directed so that the direction of interaction is important for the functioning of the systems. Recently, it has been revealed that directed networks have richer structures such as directed assortativity (Foster et al., 2010) and flow hierarchy (Mones, 2013).

In our previous work, we proposed a new path notion involving directedness called lateral path that can be seen as the dual notion to the usual directed path (Haruna, 2011). Based on category theoretic formulation, we derived the lateral path as a natural path notion associated with the dynamic mode of biological networks: a network is a pattern constructed by gluing functions of entities constituting the network (Haruna, 2012). Thus, its functionality is coherence, whereas the functionality of the directed path is transport. We showed that there is a division of labor with respect to the two functionalities within a network for several types of biological networks: gene regulation, neuronal and ecological ones (Haruna, 2012). It was suggested that the two

complementary functionalities are realized in biological systems by making use of the two ways of tracing on a directed network, namely, lateral and directed.

In this paper, we address robustness of ecological flow networks with respect to the lateral path and directed path, respectively. Since the natural connectedness notion associated with the directed path is the strong connectedness, we consider robustness of the giant strongly connected component (GSCC) for the latter. For the former, robustness of the giant laterally connected component (GLCC) is of interest. Thus, we assess robustness of ecological flow networks in terms of two different functionalities, namely, coherence and circulation, both of which are important for the functioning of them (Ulanowicz, 1997).

Robustness of ecological networks is an intriguing issue in recent studies (Montoya et al., 2006; Bascompte, 2009). Initially, robustness of general complex networks has been argued qualitatively in terms of critical thresholds for the existence of the giant component (Albert et al., 2000; Cohen et al., 2001). For ecological networks, their robustness has been measured by the size of secondary extinctions (Solé and Montoya, 2001; Dunne et al., 2002). Here, we employ a recently proposed idea to measure robustness quantitatively (Schneider et al., 2011; Herrmann et al., 2011). As a first step, we consider only random failure of arcs. The size of giant components is measured by the number of arcs involved because laterally connected components are defined only on the set of arcs.

Here, we study the impact of realistic network structures on robustness with respect to the two functionalities. Two complementary measures of it are proposed by comparing the robustness of a given real network with that of the two null models: random networks with the same degree distribution and those with the same average degree. The robustness of the two null models is calculated by theoretically solving the percolation problem on the configuration model, random networks with an arbitrary degree distribution (Newman et al., 2001).

This paper is organized as follows. In Section 2, we develop a theory to calculate the size of GLCC and GSCC under

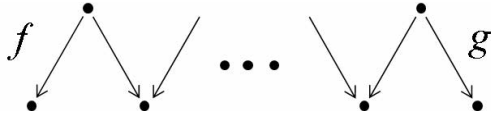


Figure 1: An example of lateral path.

random removal of arcs in the configuration model. In Section 3, we propose two measure for the impact of realistic structures on robustness of networks by using the theoretical result obtained in Section 2. In Section 4, the proposed measures are applied to 10 ecological flow networks. In Section 5, we discuss the results and indicate future directions.

### Random Removal of Arcs in the Configuration Model

In this section, we consider a percolation problem, random removal of arcs, in the configuration model with respect to the lateral connectedness and the strong connectedness.

A *lateral path* in a directed network is a path in the network such that the direction of arcs involved changes alternately (Haruna, 2012) (Fig. 1). Two arcs are called *laterally connected* if they are connected by a lateral path (Haruna, 2011). Lateral connectedness defines an equivalence relation on the set of arcs. Each equivalence class is called *laterally connected component*.

Since lateral connectedness is defined on the set of arcs, here we also consider strong connectedness for arcs. Two arcs are called *strongly connected* if there is a directed path from one arc to the other arc, and vice versa.

Let us consider a random directed network with degree distribution  $P(k_i, k_o)$ .  $P(k_i, k_o)$  is the fraction of nodes in the network with in-degree  $k_i$  and out-degree  $k_o$ . We make use of the generating function formalism (Callaway et al., 2000; Newman et al., 2001) to calculate the sizes of giant laterally or strongly connected components (in short, GLC-C or GSCC, respectively) after removing arcs uniformly at random with probability  $1 - \phi$ , where  $\phi$  is the occupation probability.

The generating function for  $P(k_i, k_o)$  is

$$G(x, y) = \sum_{k_i, k_o} P(k_i, k_o) x^{k_i} y^{k_o}. \quad (1)$$

The average degree  $z := \langle k_i \rangle = \langle k_o \rangle$  is given by

$$z = \frac{\partial G}{\partial x}(1, 1) = \frac{\partial G}{\partial y}(1, 1). \quad (2)$$

Let  $P_i(k_i) := \sum_{k_o} P(k_i, k_o)$  be the in-degree distribution and  $P_o(k_o) := \sum_{k_i} P(k_i, k_o)$  the out-degree distribution. Their generating functions are

$$F_0(x) := G(x, 1) \text{ and } H_0(y) := G(1, y), \quad (3)$$

respectively.

We introduce four excess degree distributions and corresponding generating functions that are necessary for the calculation in what follows.

First, let  $P_0(k)$  be the probability that the number of the other arcs arriving at the target node of a randomly chosen arc is  $k$  (Fig. 2 (a)). It is given by

$$P_0(k) := \frac{1}{z} \sum_{k_0} (k+1) P(k+1, k_0) \quad (4)$$

and its generating function is

$$F_{1,0}(x) := \sum_k P_0(k) x^k = \frac{1}{z} \frac{\partial G}{\partial x}(x, 1) = \frac{1}{z} \frac{\partial F_0}{\partial x}(x). \quad (5)$$

Second, let  $P_1(k)$  be the probability that the number of arcs arriving at the source node of a randomly chosen arc is  $k$  (Fig. 2 (b)). It is given by

$$P_1(k) := \frac{1}{z} \sum_{k_0} k_0 P(k, k_0) \quad (6)$$

and its generating function is

$$F_{1,1}(x) := \sum_k P_1(k) x^k = \frac{1}{z} \frac{\partial G}{\partial y}(x, 1). \quad (7)$$

Third, let  $Q_0(k)$  be the probability that the number of the other arcs leaving from the source node of a randomly chosen arc is  $k$  (Fig. 2 (c)). It is given by

$$Q_0(k) := \frac{1}{z} \sum_{k_i} (k+1) P(k_i, k+1) \quad (8)$$

and its generating function is

$$H_{1,0}(y) := \sum_k Q_0(k) y^k = \frac{1}{z} \frac{\partial G}{\partial y}(1, y) = \frac{1}{z} \frac{\partial H_0}{\partial y}(y). \quad (9)$$

Finally, let  $Q_1(k)$  be the probability that the number of arcs leaving from the target node of a randomly chosen arc is  $k$  (Fig. 2 (d)). It is given by

$$Q_1(k) := \frac{1}{z} \sum_{k_i} k_i P(k_i, k) \quad (10)$$

and its generating function is

$$H_{1,1}(y) := \sum_k Q_1(k) y^k = \frac{1}{z} \frac{\partial G}{\partial x}(1, y). \quad (11)$$

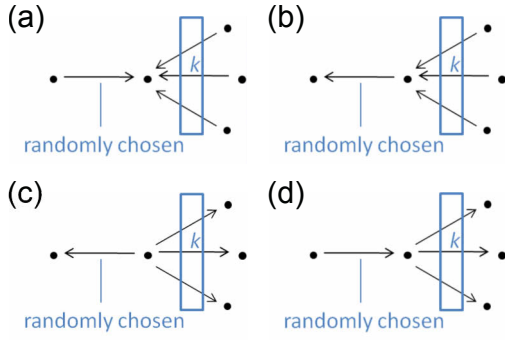


Figure 2: Four excess degree distributions. See the main text for details.

### Giant Laterally Connected Component

Let  $u$  be the average probability that an arc is not connected to the GLCC via a particular arc with the same target and  $v$  the average probability that an arc is not connected to the GLCC via a particular arc with the same source. Then, the average probability that an occupied arc does not belong to the GLCC is

$$\sum_{k,l} P_0(k) u^k Q_0(l) v^l = F_{1,0}(u) H_{1,0}(v). \quad (12)$$

Hence, the size of the GLCC is

$$L = \phi(1 - F_{1,0}(u) H_{1,0}(v)). \quad (13)$$

The values of  $u$  and  $v$  are calculated by the following set of equations:

$$\begin{cases} u = \sum_k Q_0(k)(1 - \phi + \phi v^k) = (1 - \phi) + \phi H_{1,0}(v) \\ v = \sum_k P_0(k)(1 - \phi + \phi u^k) = (1 - \phi) + \phi F_{1,0}(u). \end{cases} \quad (14)$$

The critical occupation probability  $\phi_{L,c}$  for the appearance of GLCC can be obtained from the linear stability analysis of the trivial solution  $(u, v) = (1, 1)$  of (14). It turns out to be

$$\phi_{L,c} = \frac{z}{\sqrt{(\langle k_i^2 \rangle - z)(\langle k_o^2 \rangle - z)}}. \quad (15)$$

### Giant Strongly Connected Component

The calculation of the size of the GSCC is similar to the node component case (Dorogovtsev et al., 2001; Schwartz et al., 2002). In (Serrano and De Los Rios, 2007), five notions of edge components are considered. For our purpose, consideration on the usual three components (in-, out- and strongly connected) as in the node component case are enough. However, these are implicit in the following calculation.

Let  $u$  be the average probability that an arc is not connected to the GSCC via a particular arc leaving from its target and  $v$  the average probability that an arc is not connected to

the GSCC via a particular arc arriving at its source. Then, the average probability that an occupied arc does belong to the GSCC is

$$\sum_{k,l} Q_1(k)(1 - u^k) P_1(l)(1 - v^l) = (1 - H_{1,1}(u))(1 - F_{1,1}(v)). \quad (16)$$

Hence, the size of the GSCC is

$$S = \phi(1 - H_{1,1}(u))(1 - F_{1,1}(v)). \quad (17)$$

The values of  $u$  and  $v$  are calculated by the following set of equations:

$$\begin{cases} u = \sum_k Q_1(k)(1 - \phi + \phi u^k) = (1 - \phi) + \phi H_{1,1}(u) \\ v = \sum_k P_1(k)(1 - \phi + \phi v^k) = (1 - \phi) + \phi F_{1,1}(v). \end{cases} \quad (18)$$

The critical occupation probability  $\phi_{S,c}$  for the appearance of GSCC is given by

$$\phi_{S,c} = \frac{z}{\langle k_i k_o \rangle}, \quad (19)$$

which is the same as in the node component case (Schwartz et al., 2002).

### Examples

We calculate the sizes of the GLCC and the GSCC as functions of the occupation probability  $\phi$  for three degree distributions: (a) Uncorrelated Poisson distribution (UPD)

$$P(k_i, k_o) = \frac{e^{-2\lambda} \lambda^{k_i + k_o}}{k_i! k_o!}, \quad (20)$$

(b) Uncorrelated exponential distribution (UED)

$$P(k_i, k_o) = \left(1 - e^{-1/\kappa}\right)^2 e^{-\frac{k_i + k_o}{\kappa}}, \quad (21)$$

and (c) Correlated Poisson distribution (CPD)

$$P(k_i, k_o) = \frac{e^{-\lambda} \lambda^{k_i}}{k_i!} \delta_{k_i, k_o}, \quad (22)$$

where  $\lambda, \kappa > 0$  are parameters and  $\delta_{k_i, k_o}$  is the Kronecker delta. The results are compared with numerical simulations in Fig. 3, which shows that the agreement between simulation and theory is well.

For critical occupation probabilities, we have  $\phi_{L,c} = \phi_{S,c} = 1/\lambda$  for UPD,  $\phi_{L,c} = (e^{1/\kappa} - 1)/2 < (e^{1/\kappa} - 1) = \phi_{S,c}$  for UED and  $\phi_{L,c} = 1/\lambda > 1/(\lambda + 1) = \phi_{S,c}$  for CPD. Thus, these examples also show that all possibilities  $\phi_{L,c} = \phi_{S,c}$ ,  $\phi_{L,c} > \phi_{S,c}$  and  $\phi_{L,c} < \phi_{S,c}$  actually occur.

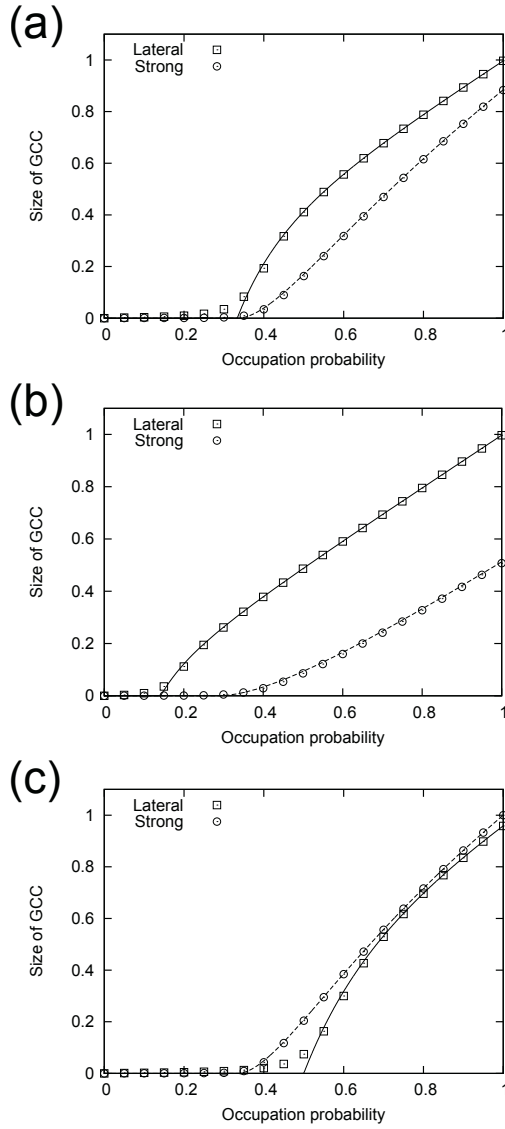


Figure 3:  $L(\phi)$  and  $S(\phi)$  for (a) the uncorrelated Poisson distribution with  $\lambda = 3$ , (b) the uncorrelated exponential distribution with  $\kappa = 4$  and (c) the correlated Poisson distribution with  $\lambda = 2$ . Lines are theoretically obtained. For (a) and (c), (14) and (18) are numerically solved. For (b), we obtain analytic expressions. Squares and circles are numerical simulations and averaged over 1000 different random removal sequences on different configuration model networks with the number of nodes 500 for (a) and (b), and 1000 for (c).

## Two Measures for Impact of Realistic Structures on Robustness

### Robustness

Given a directed network, let  $L(\phi)$  be the size of the GLCC and  $S(\phi)$  the size of the GSCC for occupation probability  $\phi$ .

Motivated by the robustness measure proposed in (Schneider et al., 2011; Herrmann et al., 2011), we define the robustness of the GLCC and that of the GSCC by

$$R_L = \int_0^1 L(\phi) d\phi \text{ and } R_S = \int_0^1 S(\phi) d\phi, \quad (23)$$

respectively.

Our robustness measure is similar to link robustness in (Zeng and Liu, 2012), however, since we measure the size of a component by the number of arcs belonging to it, it is different from link robustness. In particular, since  $L(\phi)$  and  $S(\phi)$  cannot exceed the diagonal line, we have  $R_L, R_S \leq 0.5$ .

### Gain

Given a directed network, we would like to consider how much its robustness (of the GLCC or the GSCC) is enhanced or degraded compared to a reference network. One measure is the ratio of the robustness of the given network to that of the reference network (Schneider et al., 2011). We call this measure *robustness gain*. If we denote the robustness of the given network by  $R_{given}$  and that of the reference network by  $R_{ref}$ , then the robustness gain is defined by

$$G_{given/ref} := R_{given}/R_{ref}. \quad (24)$$

We here consider three combinations of given-reference pairs: (given,ref)=(real, config), (given,ref)=(config, Poisson) and (given,ref)=(real,Poisson), where ‘real’ indicates a real-world network, ‘config’ the configuration model network with the same degree distribution and ‘Poisson’ the (uncorrelated) Poissonian network with the same average degree. The robustness gains for the three given-reference pairs are denoted by  $G_{r/c}$ ,  $G_{c/p}$  and  $G_{r/p}$ , respectively. Note that  $G_{r/p} = G_{r/c}G_{c/p}$ .

### Complement Ratio

The other way to measure the effect of realistic structures on robustness is to evaluate the amount of unrealized robustness of the reference network (namely,  $0.5 - R$ ) utilized by the given network. We define the *robustness complement ratio* for the above three combinations of given-reference pairs by

$$C_{given/ref} := \frac{R_{given} - R_{ref}}{0.5 - R_{ref}}, \quad (25)$$

where  $(given, ref) = (r, c)$ ,  $(c, p)$  or  $(r, p)$ .

Both  $G_{given/ref}$  and  $C_{given/ref}$  are considered for the lateral connectedness and the strong connectedness in next section. We write  $G_{L,given/ref}$  and  $C_{L,given/ref}$  for the former and  $G_{S,given/ref}$  and  $C_{S,given/ref}$  for the latter.

## Ecological Flow Networks

In this section, we apply the indexes introduced in previous section to relatively large 10 networks (with the number of

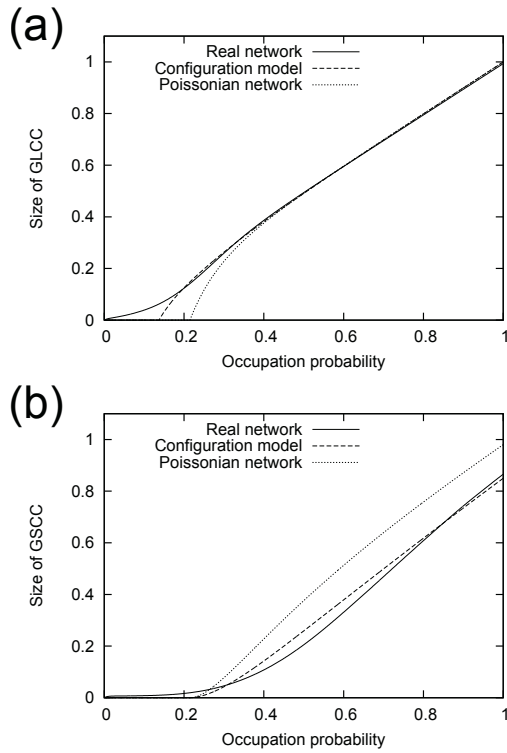


Figure 4: (a)  $L(\phi)$  and (b)  $S(\phi)$  for (vii) Middle Chesapeake Bay in Summer network (solid lines), those for the configuration model network with the same degree distribution (dashed lines) and those for the Poissonian network with the same average degree (dotted lines).

arcs  $> 100$ ) among 48 flow networks collected by R. Ulanowicz. Data are downloaded from <http://www.cbl.umces.edu/~ulan/ntwk/network.html>.

## Data

Here, we list the 10 ecological flow networks we analyze. In the following,  $N$  is the number of nodes and  $A$  is the number of arcs included in the largest weakly connected component.  $z = \langle k_i \rangle = \langle k_o \rangle$  is the average degree. The number associated to each network is the web number in the original data source. In every network, each arc indicates the existence of carbon flow from its source to target. (i) Chesapeake Bay Mesohaline Network ( $N = 26, A = 122, z = 3.4$ , Web 34). (ii) Everglades Graminoids Wet Season ( $N = 66, A = 793, z = 12.0$ , Web 40). (iii) Final Narragansett Bay Model ( $N = 32, A = 158, z = 4.9$ , Web 42). (iv) Florida Bay Wet Season ( $N = 125, A = 1938, z = 15.5$ , Web 38). (v) Lake Michigan Control Network ( $N = 34, A = 172, z = 5.1$ , Web 47). (vi) Lower Chesapeake Bay in Summer ( $N = 29, A = 115, z = 4.0$ , Web 46). (vii) Middle Chesapeake Bay in Summer ( $N = 32, A = 149, z = 4.7$ , Web

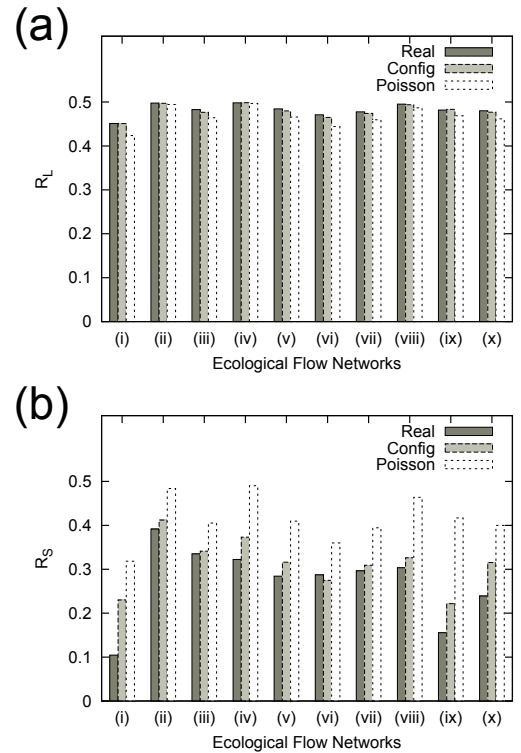


Figure 5: Robustness of (a) the GLCC and (b) the GSCC for the 10 ecological flow networks. Real: original networks, Config: the configuration model networks with the same degree distribution and Poisson: the Poissonian networks with the same average degree.

45). (viii) Mondego Estuary - Zostrea Site ( $N = 43, A = 348, z = 8.1$ , Web 41). (ix) St Marks River (Florida) Estuary ( $N = 51, A = 270, z = 5.3$ , Web 43). (x) Upper Chesapeake Bay in Summer ( $N = 33, A = 158, z = 4.8$ , Web 44).

## Results

We plot  $L(\phi)$  (Fig. 4 (a)) and  $S(\phi)$  (Fig. 4 (b)) for (vii) Middle Chesapeake Bay in Summer network, the configuration model network with the same degree distribution and the Poissonian network with the same average degree, as a typical example.  $L(\phi)$  and  $S(\phi)$  for real ecological flow networks are calculated by averaging the size of the largest connected components over 1000 random removal sequences of arcs.

The robustness values for all 10 networks are shown in Fig. 5. One can see opposite tendency on how realistic structures influence robustness between the GLCC and the GSCC.  $R_L$  tends to increase as more realistic structures are imposed on one hand,  $R_S$  tends to decrease on the other hand. However, since  $R_L$  is close to 0.5 already for the Poissonian-



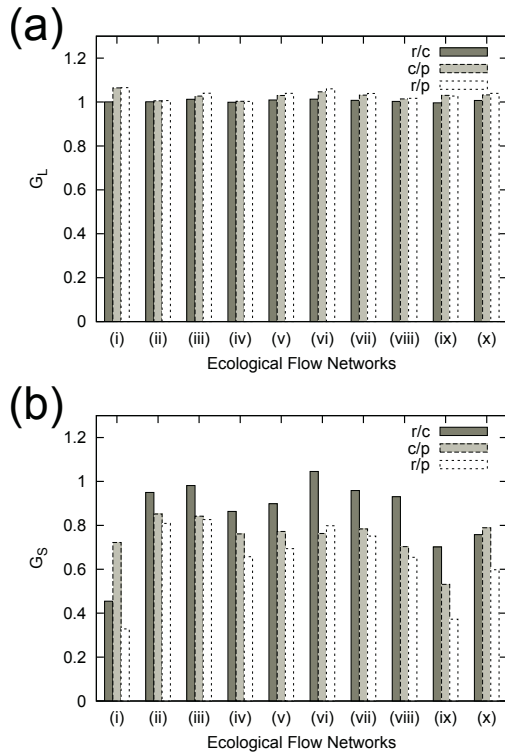


Figure 6: Robustness gain of the 10 ecological flow networks for (a) the GLCC and (b) the GSCC. Three given-reference network pairs are considered.  $r/c$ : (given,ref)=(real,config),  $c/p$ : (given,ref)=(config,Poisson) and  $r/p$ : (given,ref)=(real,Poisson). See the main text for details.

an network in most cases, the robustness gain for the GLCC is almost unity in all three given-reference pairs as seen in Fig. 6 (a). For  $R_S$ , one can see that the realistic degree distributions are the dominant factor for the degradation of robustness in most cases from Fig. 6 (b).

The tendency that realistic structures have positive impact on robustness of the GLCC can be captured more clearly by the robustness complement ratio as shown in Fig. 7. One can also see that the realistic degree distributions are the dominant factor to enhance the robustness of the GLCC in most cases.

## Discussions

Whether realistic structures of ecological networks have positive impact on their robustness or stability or not is controversial (Allesina and Tang, 2012). The answer to this question generally depends on the types of ecological interaction and dynamic processes of interest (Thébault and Fontaine, 2010; Allesina and Tang, 2012). In this paper, we focused on robustness of ecological flow networks under random failure of arcs with respect to the two different

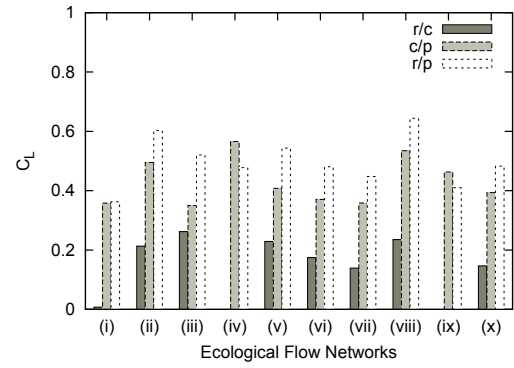


Figure 7: Robustness complement ratio of the 10 ecological flow networks for the GLCC. Three given-reference network pairs are considered.  $r/c$ : (given,ref)=(real,config),  $c/p$ : (given,ref)=(config,Poisson) and  $r/p$ : (given,ref)=(real,Poisson). Data that have negative values are omitted.  $C_S$  has negative values except one case (data not shown). See the main text for details.

functionalities, namely, coherence and circulation. The former is captured by the robustness of the GLCC and the latter by that of the GSCC. We found that they exhibit opposite tendency for constraints by the realistic network structures: the realistic network structures enhance the robustness of the GLCC on one hand, they degrade that of the GSCC on the other hand. In both case, it is suggested that the realistic degree distributions are one of the most important factors.

The former result seems to be consistent with the food-web stabilizing factor proposed in (Gross et al., 2009): “(i) species at high trophic levels feed on multiple prey species and (ii) species at intermediate trophic levels are fed upon by multiple predator species”, because such patterns in a network could contribute to make multiple lateral paths between arcs. Whereas, the latter result could provide a quantitative support for the ‘autocatalytic view’ on ecological flow networks proposed by R. Ulanowicz (Ulanowicz, 1997).

Our result in this paper suggests that complex networks can be both robust and fragile in a different sense from that in (Albert et al., 2000): under the same attack strategy, robust for one functionality and fragile for another functionality.

It is of interest whether the same tendency can be seen or not for the other various attack strategies (Holme and Kim, 2002) and for the other kinds of directed biological networks such as gene regulation and brain. Research results on these issues will be reported elsewhere near future.

## Acknowledgements

This work was partially supported by JST PRESTO program.

## References

- Albert, R., Jeong, H., and Barabási, A.-L. (2000). Error and attack tolerance of complex networks. *Nature*, 406:378–382.
- Allesina, S. and Tang, S. (2012). Stability criteria for complex ecosystems. *Nature*, 483:205–208.
- Bascompte, J. (2009). Disentangling the web of life. *Science*, 325:416–419.
- Callaway, D. S., Newman, M. E. J., Strogatz, S. H., and Watts, D. J. (2000). Network robustness and fragility: percolation on random graphs. *Phys. Rev. Lett.*, 85:5468–5471.
- Cohen, R., Erez, K., ben Avraham, D., and Havlin, S. (2001). Breakdown of the internet under intentional attack. *Phys. Rev. Lett.*, 86:3682–3685.
- Dorogovtsev, S. N., Mendes, J. F. F., and Samukhin, A. N. (2001). Giant strongly connected component of directed networks. *Phys. Rev. E*, 64:025101(R).
- Dunne, J. A., Williams, R. J., and Martinez, N. D. (2002). Network structure and biodiversity loss in food webs: robustness increases with connectance. *Ecology Letters*, 5:558–567.
- Foster, J. G., Foster, D. V., Grassberger, P., and Paczuski, M. (2010). Edge direction and the structure of networks. *Proc. Natl. Acad. Sci. USA*, 107:10815–10820.
- Gross, T., Rudolf, L., Levin, S. A., and Dieckmann, U. (2009). Generalized models reveal stabilizing factors in food webs. *Science*, 325:747–750.
- Haruna, T. (2011). Global structure of directed networks emerging from a category theoretical formulation of the idea “objects as processes, interactions as interfaces”. In Lenaerts, T., Giacobini, M., Bersini, H., Bourguine, P., Dorigo, M., and Doursat, R., editors, *Advances in Artificial Life, ECAL 2011, Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*, pages 310–317. MIT Press, Cambridge.
- Haruna, T. (2012). Theory of interface: category theory, directed networks and evolution of biological networks. arXiv:1210.6166.
- Herrmann, H. J., Schneider, C. M., Moreira, A. A., Andrade Jr, J. S., and Havlin, S. (2011). Onion-like network topology enhances robustness against malicious attacks. *J. Stat. Mech.*, P01027.
- Holme, P. and Kim, B. J. (2002). Attack vulnerability of complex networks. *Phys. Rev. E*, 65:056109.
- Mones, E. (2013). Hierarchy in directed random networks. *Phys. Rev. E*, 87:022817.
- Montoya, J. M., Pimm, S. L., and Solé, R. V. (2006). Ecological networks and their fragility. *Nature*, 442:259–264.
- Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, 45:167–256.
- Newman, M. E. J., Strogatz, S. H., and Watts, D. J. (2001). Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E*, 64:026118.
- Schneider, C. M., Moreira, A. A., Andrade Jr, J. S., Havlin, S., and Herrmann, H. J. (2011). Mitigation of malicious attacks on networks. *Proc. Natl. Acad. Sci. USA*, 108:3838–3841.
- Schwartz, N., Cohen, R., ben Avraham, D., Barabási, A.-L., and Havlin, S. (2002). Percolation in directed scale-free networks. *Phys. Rev. E*, 66:015104(R).
- Serrano, M. A. and De Los Rios, P. (2007). Interfaces and the edge percolation map of random directed networks. *Phys. Rev. E*, 76:056121.
- Solé, R. V. and Montoya, J. M. (2001). Complexity and fragility in ecological networks. *Proc. R. Soc. Lond. B*, 268:2039–2045.
- Thébault, E. and Fontaine, C. (2010). Stability of ecological communities and the architecture of mutualistic and trophic networks. *Science*, 329:853–856.
- Ulanowicz, R. E. (1997). *Ecology, the Ascendent Perspective*. Columbia University Press, New York.
- Zeng, A. and Liu, W. (2012). Enhancing network robustness against malicious attacks. *Phys. Rev. E*, 85:066130.

## Bootstrapping back the climate with self-organization

Vítor V. Vasconcelos<sup>1,2,3</sup>, Flávio L. Pinheiro<sup>1,2,3</sup>, Francisco C. Santos<sup>3,1</sup> and Jorge M. Pacheco<sup>4,5,1</sup>

<sup>1</sup>ATP-group, CMAF, Instituto para a Investigação Interdisciplinar, P-1649-003 Lisboa, Portugal

<sup>2</sup>Centro de Física da Universidade do Minho, 4710 - 057 Braga, Portugal

<sup>3</sup>INESC-ID & Instituto Superior Técnico, IST-Taguspark, 2744-016 Porto Salvo, Portugal

<sup>4</sup>Departamento de Matemática e Aplicações da Universidade do Minho, 4710 - 057 Braga, Portugal

<sup>5</sup>Centro de Biologia Molecular e Ambiental da Universidade do Minho, 4710 - 057 Braga, Portugal

vvvasconcelos@fc.ul.pt

Countries and citizens often raise significant expectations every time a new International Environmental Summit is settled. Unfortunately, few solutions have come out of these meetings. This represents a challenge on our current understanding of models on decision-making: more effective levels of discussion, agreements and coordination must become accessible (Barrett, 2005).

Mitigating the effects of climate change requires cooperation, and arguably the welfare of our planet accounts for the most important and paradigmatic example of a public good game humans face: a global good from which every single person profits, whether she contributes or not to maintain it. However, these summits failed to recognize the well-studied difficulties of cooperation in public-good games. Indeed, in most cooperation problems faced by humans, individuals, regions or nations opt to be free riders, hoping to benefit from the efforts of others while choosing not to make any effort themselves driving the population into the tragedy of the commons. When dealing with such an essential public good as climate, many efforts are made to avoid this, so that efforts are shared for all and balanced measures can then be taken.

One of the multiple flaws often appointed to such agreements is a deficit in the overall perception of risk of widespread future losses, in particular the perception of those occupying key positions in the overall political network that underlies the decision process (Santos, Santos and Pacheco, 2008; Santos and Pacheco, 2011). Another problem relates to the lack of sanctioning mechanisms to be imposed on those who do not contribute (or stop contributing) to the welfare of the planet. Moreover, agreeing on the way punishment should be implemented is far from reaching a consensus, given the difficulty in converging on the pros and cons of some procedures against others, and (occasionally) narrow impact of punishment in promoting cooperative actions (Vukov et al., 2013). The impasse over these measures is expected since their consequences do not have a solid theoretical or even experimental background.

Here we discuss *i*) the effect of group size and risk awareness in the decision making process and *ii*) the emergence and impact of different types of sanctioning in deterring non-cooperative behavior in climate agreements, as reported in (Vasconcelos, Santos and Pacheco, 2013; Santos,

Vasconcelos et al. 2012). To this end, climate agreements are defined as Collective Risk Dilemmas (**CRD**), a simple Public Goods game with uncertainty that mimetizes the problem at stake (Santos and Pacheco, 2011). We model the decision making process as a dynamical process, in which behaviours evolve in time, taking into consideration decisions and achievements of others, which influence one's own decisions. We implement such behavioural dynamics in the framework of Evolutionary Game Theory, in which the individuals are simulated to respond to the most successful (or fit) behaviours. This way, one is able to describe strategic interactions between individuals, complemented by evolutionary principles. In particular, we do so in finite populations, where such fitness driven dynamics occurs in the presence of errors (leading to stochastic effects), both in terms of errors of imitation as well as in terms of behavioral mutations ( $\mu$ ), the latter accounting for spontaneous exploration of the possible strategies. Therefore, instead of resorting to complex and rational planning or rules, individuals revise their behavior by peer-influence, creating a complex dynamics akin to many evolutionary systems.

We consider, a population of finite size  $Z$ , in which individuals engage in the aforesaid  $N$ -person dilemma. Here, each individual is able to contribute or not to a common good, i.e. to cooperate or to defect, respectively. Game participants have each an initial endowment, or benefit,  $b$ . Cooperators contribute a fraction of their endowment, the cost,  $c < b$ , while defectors do not contribute. Irrespectively of the scale at which agreements are tried, they demand a minimum number of contributors to come into practice. Hence, whenever parties fail to achieve a previously defined minimum of contributions, they may fail to achieve the goals of such agreement (which can also be understood as the benefit  $b$ ), being this outcome, in the worst possible case, associated with an appalling doomsday scenario. To encompass this feature in the model we require a minimum collective investment to ensure success: if the group of size  $N$  does not contain at least  $M$  contributors, all members will lose their remaining endowments with a probability  $(1-r)$ , the risk; otherwise, everyone will keep whatever they have. Hence,  $M < N$  represents a coordination threshold necessary to achieve a collective benefit. We obtain an unambiguous agreement with recent experiments, together with several concrete predictions:

we address the impact of risk in several configurations, from large to small groups, from deterministic towards stochastic behavioral dynamics. Overall, we show how the emerging dynamics depends heavily on the perception of risk.

We find that the impact of risk is enhanced in the presence of small behavioral mutations and errors and whenever global coordination is attempted in a majority of small groups under stringent requirements to meet co-active goals (Santos, Vasconcelos et al. 2012). This result calls for a reassessment of policies towards the promotion of public endeavors: instead of world summits, decentralized agreements between smaller groups, possibly focused on region-specific issues, where risk is high and goal achievement involves large quorums for agreement, are prone to significantly raise the probability of success in coordinating to tame the planet's climate.

We also show how individuals may effectively self-organize their actions towards cooperation, by creating community enforcement institutions that are able to punish those who row against collective interest. We present the effects of punishment via institutions when playing against defectors (which leads to higher-order cooperation dilemmas). Moreover, we offer insights on the scale at which such institutions should be implemented, providing better conditions both for cooperation to thrive and for ensuring the maintenance of such institutions (Vasconcelos, Santos and Pacheco, 2013). This result is particularly relevant whenever perception of risk of collective disaster, alone, is not enough to provide the means to achieve a cohesive configuration – in this case, sanctioning institutions may provide an escape hatch to the otherwise tragedy of the commons that humanity is falling into.

This model provides a “bottom-up” approach to the problem, in which collective cooperation is easier to achieve in a distributed way, eventually involving regions, cities, NGOs and, ultimately, all citizens. Moreover, by promoting regional or sectorial agreements, we are opening the door to the diversity of economic and political structure of all parties, which, as showed before can be beneficial to cooperation. Naturally, we are aware of the many limitations of a bare model such as this, in which the complexity of human interactions has been overlooked. From higher levels of information, to non-binary investments, additional layers of realism can be introduced in the model. Moreover, even from a modeling perspective, several extensions and complex aspects common to human socio-economical systems could be further explored. On the other hand, the simplicity of the dilemma introduced here, makes it generally applicable to other problems of collective cooperative action, which will emerge when the risks for the community are high and high-level institutions may self-organize, something that repeatedly happened throughout human history, from ancient group hunting to voluntary adoption of public health measures. In light of our results in which bottom-up approaches are clearly favored by evolution and self-organization, the widely-repeated motto “Think globally but act locally” would hardly appear more appropriate.

## References

- Vasconcelos, V. V., Santos, F.C and Pacheco, J. M. (2013). A bottom-up institutional approach to cooperation in the governance of risky commons. *Nature Climate Change*, (in print).
- Vukov, J., Pinheiro, F. L., Santos, F. C. and Pacheco, J. M. (2013) Reward from Punishment Does Not Emerge at All Costs, *PLoS Computational Biology* 9 (1) e1002868.
- Santos, F. C., Vasconcelos, V. V., Santos, M. D., Neves, P. N. B. and Pacheco, J. M. (2012). Evolutionary Dynamics of Climate Change Under Collective-Risk Dilemmas. *Mathematical Models and Methods in Applied Sciences (M3AS)*, 22, Supp1 1140004.
- Santos, F. C. and Pacheco, J. M. (2011). Risk of collective failure provides an escape from the tragedy of the commons. *Proc. Natl. Acad. Sci. USA* 108 (26) 10421-10425.
- Santos, F. C., Santos, M. D and Pacheco, J. M. (2008). Social diversity promotes the emergence of cooperation in public goods games. *Nature* 454 213-216.
- Barrett, S. (2005), *Environment and statecraft: the strategy of environmental treaty-making*. (Oxford University Press, USA).

# Neural agents can evolve to reproduce sequences of arbitrary length

Benjamin Inden<sup>1</sup> and Jürgen Jost<sup>2,3</sup>

<sup>1</sup>Artificial Intelligence Group, Bielefeld University, Bielefeld, Germany  
binden@techfak.uni-bielefeld.de

<sup>2</sup>Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany  
jost@mis.mpg.de

<sup>3</sup>Santa Fe Institute, Santa Fe, New Mexico, USA

## Abstract

We demonstrate that neural agents can evolve behavioral sequences of arbitrary length. In our framework, agents in a two-dimensional arena have to find the secure one among two possible patches, and which of them is secure changes over time. Evolution of arbitrarily long behavioral sequences is achieved by extending the neuroevolution method NEAT with two techniques: Only newly evolved network structure is subject to mutations, and inputs to the neural network are provided in an incremental fashion during evolution. It is suggested that these techniques are transferable to other neuroevolution methods and domains, and constitute a step towards achieving open-ended evolution. Furthermore, it is argued that the proposed techniques are strongly simplified models of processes that to some degree occur naturally in systems with more flexible genetic architectures.

## Introduction

Evolutionary robotics is an approach towards the design of controllers (and possibly also bodies) of robots that uses a process of artificial evolution. The controllers of the resulting robots are often in the form of artificial neural networks. It is conceivable that eventually robots with very complex goal-directed behaviors can be designed using this method. However, past work in the field has typically produced behaviors that are very simple as compared to behaviors of conventionally designed robot controllers.

Complex behaviors can arise by the interplay of environment, body, and controller even if the controller itself is not very complex (Pfeifer and Gómez, 2005). Nevertheless, within a given environment, there is obviously a correlation between the complexity of the controller and the complexity of the behavior. Therefore, it is desirable to have methods for the evolution of neural networks that can make the networks more and more complex over time.

For many of the earliest evolutionary robotics experiments, neural networks with a fixed topology, i.e., a fixed number of nodes and connections, were used (Nolfi and Floreano, 2000). That way, the achievable complexity is obviously limited. Later, a method called NEAT was introduced (Stanley and Miikkulainen, 2002). This neuroevolution method starts evolution using networks without any

hidden nodes and subsequently adds neurons and connections by carefully designed mutation operators. It has been shown that complexification during evolution does indeed occur when using NEAT, and can lead to neural networks with in the order of, say, 10 to 20 hidden nodes (Stanley and Miikkulainen, 2004). NEAT has subsequently been widely used for various evolutionary robotics experiments.

Of course, ultimately one needs more complexity than that evolved by NEAT and similar methods. Therefore, a more recent trend in neuroevolution is the development of methods that use developmental or generative encodings for neural networks, which enables them to produce large neural networks from comparatively small genomes. Examples of such methods include HyperNEAT, NEATfields and Compressed Network Complexity Search (Stanley et al., 2009; Inden et al., 2012; Gomez et al., 2012). These methods enhance the scalability of neuroevolution significantly as compared to methods using a simpler direct encoding like NEAT.

While evolution can produce comparatively complex networks with hundreds of neurons using these methods, the complexity of the behavior that can be produced by evolution (i.e., adaptive behavior, not just random behavior) is still subject to some limitations. These limitations arise from the fact that a genome consisting of  $l$  elements, each of which can be chosen among an alphabet of size  $b$ , can encode a choice between  $b^l$  different phenotypes at most. Depending on the method used, this can mean that the maximum size of the encoded network will be limited (unless it is preset by the experimenter like in standard HyperNEAT) or the maximum algorithmic complexity of its connection pattern will be limited (unless the encoding includes techniques that refer to externally or environmentally supplied sources of complexity). Even if none of those two limitations apply, the fact remains that the number of referentiable phenotypes is limited, so the complexity behavior as achieved by selection remains limited unless the genome length  $l$  is increased. However, a selection-driven unlimited increase of  $l$  is usually not possible even though the methods allow for it in principle. To see why this is so, consider what happens as more and more genes are added to a given genome. If the mutation



rate per gene is fixed, then more and more mutations will arise in one genome as it gets longer. At some point, mutation will destroy the information in the genome despite the presence of selection. In the theoretical biology literature, this point is known as error threshold (Eigen, 1971; Stadler and Stadler, 2003). However, many neuroevolution methods (among them NEAT) typically apply only one mutation per genome regardless of its size. That way, mutations will not overpower selection, but on the other hand, per gene mutation rates will decrease. Together with them, the mutation rates on individual features of the phenotype will also decrease. As we are interested in adaptive behavior of increasing complexity, which consists of more and more individual features, the waiting times for adaptive changes of or extensions to particular features will increase until evolution practically comes to a halt. One common way of describing this is saying that the search space increases exponentially with the number of variables (the variables correspond to the phenotypic features here). One might think that as this happens, new opportunities arise for evolution in the form of extradimensional bypasses (Conrad, 1990; Bongard and Paul, 2001), but it is unlikely that this effect keeps pace with the increase of the search space for fitness functions as typically used in evolutionary robotics. In fact, it is a common heuristic in neuroevolution to make the search space as small as possible by use of prior information on the task. The NEAT method also starts with the smallest possible neural network topology and only then gradually explores larger topologies just to keep the search space small.

However, there is a way for evolution to avoid this dilemma: by changing the genetic architecture such that mutations will with a higher than random probability hit the right places. Ideally, successful features of an organism would be conserved by reducing the local mutation rates, whereas features under adaptive evolution would have increased local mutation rates. Mutation rates can be changed either by changing local application rates of particular mutation operators (like in some variants of self-adaptive evolution strategies (Beyer and Schwefel, 2002)), or by changing the mutational target sizes of the respective features if the used encoding and mutation operators allow for such reorganizations of the genetic representations of those features. In fact, it has been shown in simulations using the artificial life system AVIDA that the representations of conserved features can become compressed over time just by applying normal mutation and selection (Ofria et al., 2003). In principle, these changes of genetic architecture might also arise when using methods like HyperNEAT. However, selective pressure for these kinds of changes might primarily arise indirectly through increased evolvability of the offspring (for a review of different ideas, see (Hansen, 2006)) and may therefore be too weak in many situations, or become effective only over time scales that are currently beyond the reach of artificial evolution.

In this article, we propose a more direct method to guide mutations towards features under active evolution, and away from previously evolved adaptive features. The basic idea is that only those parts of a neural network that were created by mutations most recently can be mutated. Older structures are frozen and cannot be mutated any more. The NEAT method has a feature that makes implementation of this approach very easy: every time a neuron or a connection arises by mutation, it receives a globally unique identification number. One simple implementation is just to take this number from a counter that is increased whenever a new number is assigned. In that case, the numbers also provide an indication of relative time. The genes in a genome can be ordered according to the time of their creation, and only a fixed number of the newest genes are then available for mutations.

This method is related to earlier approaches on incremental evolution. E.g., one method uses a new module for each new fitness function term, and may restrict mutations in the older modules (Pasemann et al., 2001). However, our approach is more fine-grained and gradual, and makes use of the identification numbers that are used in NEAT and derived methods like HyperNEAT and NEATfields. No manual definition of modules or partitioning of fitness functions is necessary. We expect that if used in combination with other recent neuroevolution techniques, the technique presented here will enable open-ended evolution and complexification of neural agents.

Open-ended evolution is an interesting concept from the perspective of both evolutionary robotics and theoretical biology. A precise measure for evolutionary activity that was introduced several years ago (Bedau et al., 1998) links open-ended evolution to the unbounded growth of cumulative evolutionary activity, which can be achieved by unlimited growth of diversity and/or unlimited complexification. An abstract model that has unbounded growth of diversity has been introduced shortly afterwards (Maley, 1999). A more complex artificial life system with unbounded evolutionary activity has also been designed (Channon, 2006). One of us has recently presented an abstract model of open-ended coevolution (Inden, 2012). In that model, the genotype and phenotype of an organism is a string of integer numbers from some bounded range. The fitness of an organism depends on the results of matching its number string against those of organisms from the other population. In many variants of the model, mutations are only performed at the end of the string. The present article aims at initiating a transfer of this approach, and the resulting open-ended evolution, to the domain of neural agent evolution.

## Methods

### The patches task

An agent resides in a two-dimensional arena where  $x$  and  $y$  coordinates are constrained to the range  $[-1, 1]$  each. At each time step, it can change its position by  $o_i \cdot 0.1$  in both

dimensions simultaneously, where  $o_i \in [-1, 1]$ ,  $i \in \{1, 2\}$ , is the respective neural network output. This means it can get from one border of the arena to the other in twenty time steps. Every twenty time steps, the position of the agent is checked. Only if it is on the correct nest site in its arena, it will survive into the next round of twenty time steps. The left nest site covers all positions with  $x \in [-1, -0.5]$ , whereas the right nest site covers all positions with  $x \in (0.5, 1]$ . A long binary random sequence is generated before evolution begins, and determines which nest site is the correct one in each round. This means that the agents basically have to learn to produce a binary random sequence by evolution. The agents get the following input: a bias input, their current position, and a number of inputs indicating the number of the current round (as detailed below).

For every successful round, a reward of 1.0 is added to the fitness. In the final (unsuccessful) round, the agent obtains an additional reward of  $\frac{1+pos_x}{4}$  if the right nest site was the correct target, or  $\frac{1-pos_x}{4}$  if the left nest site was the correct target.

It can be seen from this description that actually only one spatial dimension is directly relevant for the task. The  $y$  direction is introduced for purposes of visualization and to provide an additional neutral dimension for future more complicated tasks.

### The NEAT neuroevolution method

The NEAT (NeuroEvolution of Augmenting Topologies) method (Stanley and Miikkulainen, 2002) is a well known method for simultaneously evolving the topology and the connection weights of neural networks. It starts evolution with one of the simplest possible network topologies and proceeds by complexification of that topology. More specifically, the common ancestor of the whole population has one neuron for each output, each of which is connected to all inputs. There are no hidden neurons. Here, we start even simpler: Each output has one neuron, and each of this neurons is initially connected to 30% of the inputs on average. It has been shown previously that letting evolution select inputs for the neural network can result in superior performance if the input space is large as compared to starting with a fully connected network (Whiteson et al., 2005). Evolution then proceeds by adding neurons and connections.

Our NEAT implementation<sup>1</sup> uses mutation operators that are very similar to those of the original NEAT implementation for evolving the contents of the field elements. The most common operation is to choose a fraction of connection weights and either perturb them using a normal distribution with standard deviation 0.18, or (with a probability of

<sup>1</sup>To be more accurate, we use an implementation of the NEAT-fields method, which is an extension of NEAT that makes possible the evolution of large neural networks with regularities (Inden et al., 2012). However, all extension features are switched off, so the method is reduced to pure NEAT.

0.15) set them to a new value. The application probability of this weight changing operator is set to 1.0 minus the probabilities of all structural mutation operators, which amounts to 0.938 here. A structural mutation operator to connect previously unconnected neurons is used with probability 0.02, while an operator to insert neurons is used with probability 0.001. The latter inserts a new neuron between two connected neurons. The weight of the incoming connection to the new neuron is set to 1.0, while the weight of the outgoing connection keeps the original value. The idea behind this approach is to change the properties of the former connection as little as possible to minimize disruption of existing functional structures. The former connection is deactivated but retained in the genome where it might be reactivated by further mutations. There are two operators that can achieve this: one toggles the active flag of a connection and the other sets the flag to 1. They are used with probability 0.01 each.

Once a new gene arises by mutation, it receives a globally unique reference number. This number is generated by a global counter that is incremented every time a new gene arises. The innovation numbers are originally used by NEAT to align two genomes during the process of recombination (although in the experiments reported here, no recombination is used). They are also used to define a distance measure between networks, as will be explained in the next section.

The activation of the individual neurons is a weighted sum of the outputs of the neurons  $j \in J$  to which they are connected, and a sigmoid function is applied on the activation:  $o_i(t) = \tanh(\sum_{j \in J} w_{ij} o_j(t-1))$ . Like in some other NEAT implementations, connection weights are constrained to the range  $[-3, 3]$ . There is no explicit threshold value for the neurons. Instead, a constant bias input is available in all networks.

### Speciation selection

NEAT uses speciation selection by default. This method is fitness based, but uses some techniques to protect innovation that may arise during evolution against competition from fitter individuals that are already in the population. As a prerequisite, the globally unique reference numbers assigned to each gene are used to calculate a distance measure between two neural networks. The dissimilarity between two networks is calculated as  $d = c_r \#ref_c + c_w \sum \Delta w$ , where  $\#ref_c$  is the number of connections present in just one of these networks,  $\Delta w$  are the connection weight differences (summed over pairs of connections that are present in both networks), and the  $c$  variables are weighting constants with  $c_r = 1.0$ ,  $c_w = 1.0$  by default.

Using this dissimilarity measure, the population is partitioned into species by working through the list of individuals. An individual is compared to representative individuals of all species until the dissimilarity between it and a representative is below a certain threshold. It is then assigned to this species. If no compatible species is found, a new species

is created and the individual becomes its representative.

The number of offspring assigned to a species is proportional to its mean fitness. This rather weak selection pressure prevents a slightly superior species from taking over the whole population, and enables innovative yet currently inferior solutions to survive. In contrast, the selection pressure between members of the same species is much stronger: the worst 60% of the individuals belonging to that species are deleted, after which the other individuals are selected randomly. Species that have at least five individuals in the next generation also take the best individual into the next generation without mutations. If the maximum fitness of a species has not increased for more than 100 generations and it is not the species containing the best network, its mean fitness is multiplied by 0.01, which usually results in its extinction. Also, in order to keep the number of species in a specified range, the dissimilarity threshold is adjusted in every generation if necessary. Here, the initial speciation threshold is 4.0, the population size is 1000, and the target number of species is between 35 and 45. All numerical parameters for speciation selection have been taken over from previous experiments on other tasks (Inden et al., 2012).

Genetic speciation will be used as point of comparison for the experiments reported here, but the main method used is speciation based on behavioral criteria. This means that the distance between neural agents is not calculated from their genes, but from a metric of the behavior space. For the *patches* task, the final position  $(x, y)$  of an individual is recorded, and the distance between two individuals is just  $(x_1 - x_2)^2 + (y_1 - y_2)^2$ . The initial speciation threshold is set to 0.1. Everything else works as for genetic speciation.

Tournament selection with a tournament size of two and an elite of size 10 is also used for comparison.

### Incremental evolution of network architecture

Usually, mutations are applied on all genes with uniform probability. In contrast, the method introduced here allows mutations only on the  $c_m$  newest genes. The relative age of all genes is known because their innovation numbers are ordered by the time of their creation, so the smallest innovation number where mutations are still allowed can be calculated at the beginning of the mutation procedure from a list of all genes in the genome.  $c_m$  should obviously be greater than the number of genes in the common ancestors. For the *patches* task and the default input configuration described below, there are 18 genes in the common ancestor on average, and  $c_m$  is set to 25. The method is robust to some variation in this parameter.

All mutations are forbidden on older genes, including perturbations of the connection weights. However, connections between new and old neurons are allowed, as are split operations applied on connections from the input or to the output. The first exception makes connecting newly evolved structures with older structures possible, while the second opens

up possibilities for newer structure to be connected to inputs and outputs.

### Incremental provision of network input

For the task considered here, the agent needs to react differently in different rounds, therefore it needs to possess some information that is correlated to the number of the current round at any point in time. Given that neural networks can generate internal dynamics, they could be expected to track time entirely internally. One could also think of providing the current round as a binary number on several inputs, or provide some periodic inputs with different periods to simplify evolution of time-dependent behavior. Preliminary experiments have shown that all methods work to some degree, but not very well. Therefore, the approach chosen here is to provide an input for each round that is at 1.0 during that round, and at 0.0 at other times. That approach implies not only that the number of inputs is potentially unlimited, but also that it is growing linearly with the number of rounds.

In our NEAT implementation, the neuroevolution methods need to be provided with information on the input and output geometry by the task specific methods. For the purposes of the pure NEAT method as reported here, this information is just a list of network inputs and a list of network outputs<sup>2</sup>. Each input and output has a unique identification number just like the network genes have. This number is stored in the genome whenever a connection from input or to output is established by NEAT.

To evolve neural networks with a potentially unlimited number of inputs, three parameters have to be set in our approach. The first two are related to the task: The initial number of inputs  $c_s = 20$  and the input increment number  $c_i = 10$ . Again, the exact values are task specific and can be varied within reasonable bounds without much influence on the performance. In the initial generation,  $c_s$  inputs are presented in the task geometry. Whenever the task components relating to these inputs are solved sufficiently well, the next  $c_i$  inputs are added to the task geometry. For the purposes of the experiments here, a solution is sufficiently good if it survives at least  $n_i - c_i$  rounds, where  $n_i$  is the current number of network inputs provided. Once a sufficiently good individual is found, the number of provided inputs is increased for all individuals in the next generation.

The third parameter  $c_e = 25$  is the number of newest inputs that are considered by the mutation operator for establishing new connections. Connections to older inputs will not be established. Of course, some inputs could be made exempt from this rule if they are considered to be important for later stages of evolution as well. But this does not seem to be necessary for the task considered here.

<sup>2</sup>For the NEATfields method, the task geometry contains more information related to how the inputs and outputs are structured into fields. Details can be found in Inden et al. (2012).

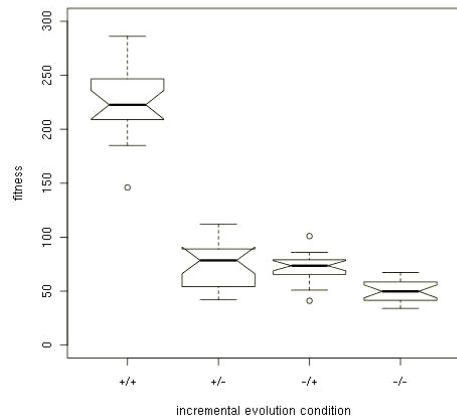


Figure 1: Performance of different configurations. The configurations are denoted by combinations such as “+/+”, where the first symbol indicates the presence of the incremental network evolution technique, and the last symbol indicates the presence of the incremental input provision technique.

## Experiments and results

The first set of experiments is designed to show that the two techniques proposed here — incremental evolution of network architecture and incremental provision of network input — do indeed lead to evolutionary learning of sequences of arbitrary length if applied to the *patches* task. NEAT with both techniques is compared against NEAT with only one or none of these techniques. As Fig. 1 shows, using both techniques leads to significantly more performance than using one technique alone, which in turn is significantly better than using none of these techniques. When using both techniques, the highest fitness reached is 227.0 on average. (As for all other experiments reported here, 20 runs of 5000 generations each have been done, and significance has been established using Wilcoxon’s rank sum test in addition to estimating it from the diagrams.) More importantly, as Fig. 2 shows, evolutionary dynamics is fundamentally different for the different configurations: When using both techniques, approximately linear growth of fitness (and of behavioral complexity by implication) with a steep slope occurs. If only the network is evolved incrementally, growth is much slower, although it may be linear as well in the depicted range. If only the inputs are provided incrementally, there is fast initial growth but later convergence. If none of the techniques is used, convergence is reached at even lower values.

It is also instructive to look at the neural network resulting from the run that achieved the highest fitness (Fig. 5). This particular network is encoded by 410 genes and consists of 39 neurons and 321 active connections (On average,

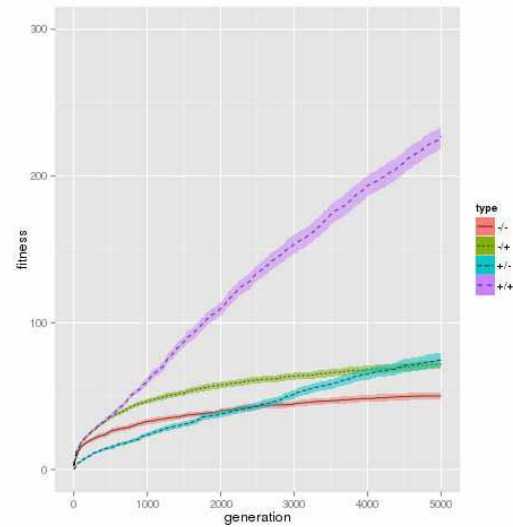


Figure 2: Mean highest fitness over the course of evolution for four different configurations denoted as in Fig. 1. The ribbons around the lines show the uncertainty in the mean (standard error).

the champions of all runs had  $36.1 \pm 1.3$  neurons). It can be seen that the output for movement in the  $x$  direction is integrated at neuron 0, whereas many different neurons directly connect to the output responsible for movement in the neutral  $y$  direction. Not all inputs are connected to the network. For those that are, there is a rather regular pattern of radial spikes leading to the hidden layer of the network. However, some hidden neurons are also connected to a few older inputs. In general, the older neurons have a higher degree of connectedness. This pattern of connectivity can be interpreted as arising from a rather regular incremental evolution of structures together with some reuse of older structure.

When knocking out a single neuron at a time, the performance of the network is decreased for 23 (59%) of the neurons. This is a lower bound for the number of functional neurons in the network (more neurons could perhaps be shown to contribute to the function in experiments where multiple neurons are knocked out simultaneously). Furthermore, if neurons are ordered according to their time of creation by a mutation, an interesting pattern emerges (Fig. 3): The order of their creation is strongly correlated to the performance of the knockout network. A plausible explanation is that neurons that evolve later typically do not affect the behavior of the agents in the early rounds of the *patches* task. The structure that controls this behavior has been frozen. The specific task of those later neurons is to control behavior in the later rounds of the task.

A second set of experiments has been designed to examine the influence of different selection methods on the per-



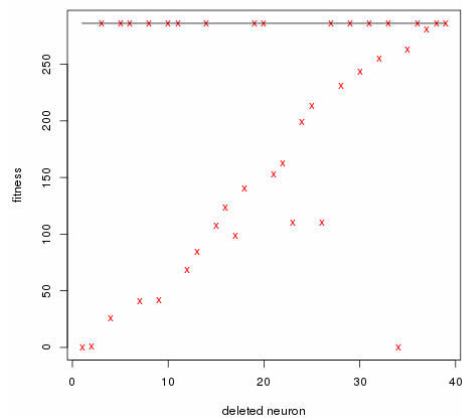


Figure 3: Fitness of an evolved network after knockout of a single neuron each. Neurons are ordered according to the time of their evolutionary emergence. The line at the top indicates the fitness of the unaltered network.

formance. Earlier experiments have shown that solutions of sufficient quality for many tasks can only be found using particular selection methods like speciation selection or methods using novelty search (Stanley and Miikkulainen, 2002; Lehman and Stanley, 2011; Inden et al., 2013). Therefore, interesting results on representations and mutation operators could be masked if inappropriate selection methods are used. We have therefore adopted the general strategy of never just considering one aspect of neuroevolution in isolation. However, as Fig. 4 shows, performance does not differ significantly between behavior speciation and tournament selection for this task. Genetic speciation performs significantly worse than the other two methods.

## Discussion

We have claimed that neural agents can learn behavioral sequences of arbitrary length using the techniques presented here, and that this represents a step towards open-ended evolution and complexification. Evidence has been provided in the form of approximately linear growth of behavioral complexity over 5000 generations. To our knowledge, the resulting adaptive behavioral complexity is a substantial advance over previous evolutionary robotics results, where agents were trained to perform a few sequential actions only.

However, it might be argued that there is nothing to guarantee further linear growth beyond what has been empirically shown, and that there even may be some inherent limitations in the method that might eventually (though much later than usual) make evolution come to a halt. To these kinds of arguments, two replies can be made: Firstly, we may (as discussed by previous authors (Bedau et al., 1998;

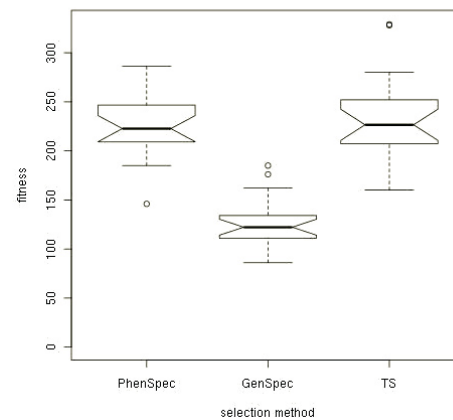


Figure 4: Performance of different selection methods for the *patches* task. PhenSpec, behavioral speciation; GenSpec, genetic speciation; TS, tournament selection.

Maley, 1999)) take a rather pragmatic approach to open-endedness. Eventually, evolution will always come to a halt because of resource limits in its environment. If we have a method that seems likely to be able to reach these resource limits provided that parameters are set appropriately, this may already qualify as leading towards open-ended evolution. Secondly, we have only described a simple implementation of our ideas because this was sufficient for the simple task studied here. More sophisticated implementations might remove some remaining limitations in the method. For example, as implemented currently, it is possible to connect new neurons to arbitrary old neurons. As the number of old neurons grows by complexification, the waiting times for connections to specific neurons again increase, which may lead to a slowdown in evolution. However, one could also easily set a second age threshold and only connect to old neurons that are younger than this second threshold. Similar solutions should be possible for other mutation operators.

Here, we let the agents only learn random sequences. When their length increases, so does their algorithmic complexity, as that complexity essentially measures the randomness of a sequence (Li and Vitányi, 1997). A fundamental question around the issue of open-ended evolution, however, is whether also the complexity concerned with the structural regularities can be made to increase by such a scheme (Ay et al., 2011). This is not yet addressed by our simulations. It would clearly be interesting to provide more regular sequences and find out how well some kind of generalization can be learned on top of this sequential learning of individual instances. We think that the ideas of incremental genome growth and restriction of mutations to the most recently evolved genes can also contribute to that issue.



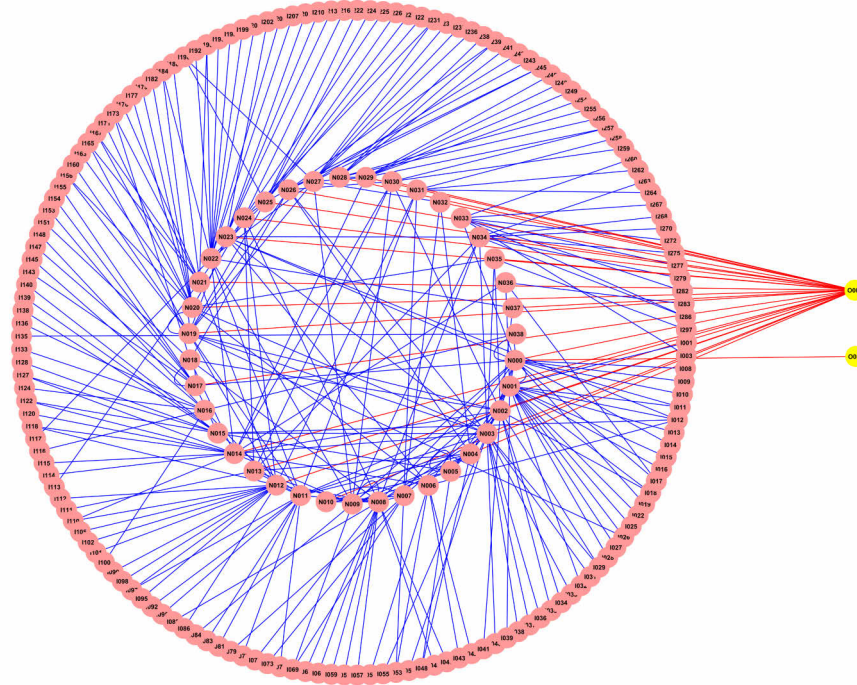


Figure 5: An evolved neural network that survives for 286 rounds. Inputs are displayed in the outer circle, hidden neurons in the inner circle, and outputs on the right. Connections to the output are displayed in red. Inputs and hidden neurons are in evolutionary order starting approximately at 3 o'clock and rotating clockwise from there.

One might ask to what extent open-ended complexification can be achieved in other domains using these methods. This is a question for further research. A first observation is that in the task studied here, individual parts of the task are presented sequentially (i.e., the agent only has to make a correct guess at some point in time once it has evolved to make correct guesses at all times before that time). This corresponds well with the incremental evolution of network structure enforced by our technique. It is unknown whether and under what conditions evolution can find a path for itself if the fitness function is less structured. It will also be interesting to see how well the methods work for real robots operating in more complex and noisy environments.

We also note that the presented techniques can be used with other neuroevolution methods that use unique identification numbers for genes. In fact, we have already implemented it for NEATfields and aim to combine incremental evolution with learning geometric regularities as is possible with NEATfields and some other methods.

Finally, while the presented technique might seem to be a very contrived addition to evolution, it could be argued that it is a strong simplification and an extreme case of something that does happen naturally to some degree in evolution if evolution occurs on complex genetic architectures. We have already mentioned that selection for compressed representa-

tions of individual features has been observed in an artificial life system (Ofria et al., 2003). It is also known that different regions in animal genomes are subject to different mutation rates, and that this is under genetic control (Martincorena et al., 2012). Evolution of genetic architectures has been the subject of intense research in recent decades (Hansen, 2006). Therefore, the ideas presented here for open-ended evolution and complexification may ultimately be relevant for a class of systems much wider than just neural agents.

**Acknowledgments** B. I. is currently funded by the Deutsche Forschungsgemeinschaft (DFG) in the Collaborative Research Center 673. Additional funding for the project “Coevolutionary complexification of autonomous agents” was supplied by Bielefeld University within the programme “Bielefelder Nachwuchsfonds”.

## References

- Ay, N., Olbrich, E., Bertschinger, N., and Jost, J. (2011). A geometric approach to complexity. *Chaos*, 21:037103.
- Bedau, M. A., Snyder, E., and Packard, N. H. (1998). A classification of long-term evolutionary dynamics. In et al., C. A., editor, *Artificial Life VI*.
- Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strate-

- gies — a comprehensive introduction. *Natural Computing*, 1:3–52.
- Bongard, J. and Paul, C. (2001). Making evolution an offer it can't refuse: Morphology and the extradimensional bypass. In *Proceedings of the Sixth European Conference on Artificial Life*.
- Channon, A. (2006). Unbounded evolutionary dynamics in a system of agents that actively process and transform their environment. *Genetic Programming and Evolvable Machines*, 7:253–281.
- Conrad, M. (1990). The geometry of evolution. *BioSystems*, 24:61–81.
- Eigen, M. (1971). Selforganization of matter and evolution of biological macromolecules. *Naturwissenschaften*, 58:465–523.
- Gomez, F., Koutník, J., and Schmidhuber, J. (2012). Compressed network complexity search. In *Proceedings of the 12th International Conference on Parallel Problem Solving from Nature*.
- Hansen, T. (2006). The evolution of genetic architecture. *Annual Reviews of Ecology, Evolution and Systematics*, 37:123–157.
- Inden, B. (2012). Open-ended coevolution and the emergence of complex irreducible functional units in iterated number sequence games. In *Proceedings of the 14th annual conference on genetic and evolutionary computation*.
- Inden, B., Jin, Y., Haschke, R., and Ritter, H. (2012). Evolving neural fields for problems with large input and output spaces. *Neural Networks*, 28:24–39.
- Inden, B., Jin, Y., Haschke, R., and Ritter, H. (2013). An examination of different fitness and novelty based selection methods for the evolution of neural networks. *Soft Computing*, 17:753–767.
- Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19:189–223.
- Li, M. and Vitányi, P. (1997). *An introduction to Kolmogorov complexity and its applications*. Springer.
- Maley, C. C. (1999). Four steps toward open-ended evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*.
- Martincorena, I., Seshasayee, A. S. N., and Luscombe, N. M. (2012). Evidence of non-random mutation rates suggests an evolutionary risk management strategy. *Nature*, 485:95–98.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics — The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press.
- Ofria, C., Adami, C., and Collier, T. C. (2003). Selective pressures on genomes in molecular evolution. *Journal of Theoretical Biology*, 222:477–483.
- Pasemann, F., Steinmetz, U., Hülse, M., and Lara, B. (2001). Robot control and the evolution of modular neurodynamics. *Theory in Biosciences*, 120:311–326.
- Pfeifer, R. and Gómez, G. (2005). Interacting with the real world: design principles for intelligent systems. *Artificial Life and Robotics*, 9:1–6.
- Stadler, B. M. R. and Stadler, P. F. (2003). Molecular replicator dynamics. *Advances in Complex Systems*, 6:47–77.
- Stanley, K. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127.
- Stanley, K. O., D'Ambrosio, D. B., and Gauci, J. (2009). A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15:185–212.
- Stanley, K. O. and Miikkulainen, R. (2004). Competitive coevolution through coevolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100.
- Whiteson, S., Stone, P., Stanley, K. O., Miikkulainen, R., and Kohl, N. (2005). Automatic feature selection in neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*.

# Spatial Organisation of Cooperation with Contingent Agent Migration

Pierre Buesser and Marco Tomassini

HEC Faculty, Information Systems Department, University of Lausanne, Switzerland

## Abstract

In the framework of game theory and cooperation, we study standard two-person population games when agents in the population are allowed to move to better positions in a two-dimensional diluted grid. We show that cooperation may thrive for small interaction radius and when mobility is low. Furthermore, we show that, even when the agents cannot change their game strategy, interesting spatial patterns do emerge as players explore their neighborhood in order to find a better place to migrate to. In the Prisoner's Dilemma and Stag-Hunt games, when the losses experienced by cooperators against defectors as well as the game and migration radius are large enough, players move in a coherent way because clusters of cooperators followed by defectors form. On the other hand, in the Hawk-Dove game or when the migration radius is small, players end up blocked into stationary clusters.

## Introduction

Systems whose parts are contained in physical space are very important in biological and social sciences since most interactions among living beings or artificial actors take place in such a space. Thus game-theoretical interactions among spatially embedded agents distributed according to a fixed structure in the plane have been studied in detail since the pioneering works of Axelrod (Axelrod, 1984) and Nowak and May (Nowak and May, 1992). The related literature is very large; see, for instance, the review article by Nowak and Sigmund (Nowak and Sigmund, 2000) and references therein for a synthesis. Most of this work was based on populations of agents arranged according to planar regular grids for mathematical simplicity and ease of numerical simulation. Recently, some extensions to more general spatial networks have been discussed in (Buesser and Tomassini, 2012). Strategic behavior on fixed spatial structures is important but, in the majority of real situations both in biology and in human societies, actors have the possibility to move around in space. Many examples can be found in biological and ecological sciences, in human populations, and in engineered systems such as ad hoc networks of mobile communicating devices or robot teams. Mobility may have positive or negative effects on cooperation, depending on several

factors. For instance, early on Enquist and Leimar (Enquist and Leimar, 1993) studied a model in which space is not explicitly represented but assortment of strategies is made non-uniform by introducing the possibility of abandoning a non-profitable relationship and searching for another partner, thus modifying the homogeneous well-mixed original population structure. Their main conclusion was that mobility may seriously restrict the evolution of cooperation. In the last decade there have been several new studies of the influence of mobility on the behavior of various games in spatial environments representing essentially two strands of research: one in which the movement of agents is seen as a random walk, and a second one in which movement may contain random elements but it is purposeful, or strategy-driven. In the present study we focus on situations where, instead of randomly diffusing, agents possess some basic cognitive abilities and they actively seek to improve their situation by moving in space represented as a discrete grid in which part of the available sites are empty and can thus be the target of the displacement. This approach has been followed, for example, in (Helbing and Yu, 2009, 2008; Jiang et al., 2010; Chen et al., 2011; Aktipis, 2004). The mechanisms invoked range from success-driven migration (Helbing and Yu, 2009), adaptive migration (Jiang et al., 2010), flocking behavior (Chen et al., 2011), and cooperators walking away from defectors (Aktipis, 2004). The general qualitative message of this work is that purposeful contingent movement may lead to highly cooperating stable or quasi-stable population states if some conditions are satisfied. Despite all the above work, the quantitative results strongly depend on the assumptions made and on the details of the models.

Our approach here is inspired by the work of (Helbing and Yu, 2008, 2009) which they call "success-driven migration" and which has been shown to be able to produce highly cooperative states. In this model, locally interacting agents playing either defection or cooperation in a two-person Prisoner's Dilemma are initially randomly distributed on a grid in equal proportions with a certain density such that there are empty grid points. Agents are updated one at a time.

When chosen for updating, the agent evaluates the current payoff she would accumulate by playing two-person games with all her current neighbors but she can also “explore” an extended square neighborhood by testing all the empty positions up to a given distance. If the player finds that it would be more profitable to move to one of these positions then she does it, choosing the best one among those tested, otherwise she stays at her current place. Helbing and Yu find that robust cooperation states may be reached by this single mechanism, even in the presence of random noise in the form of random strategy mutations and random agent relocation. Our study builds upon this work in several ways. In the first place, whilst Helbing and Yu had a single game neighborhood, we systematically investigate game neighborhood and migration neighborhood, showing that only some values of this pair of parameters allow the evolution of cooperation using success-driven migration. Second, we present systematical results for a whole game phase space including the Hawk-Dove class of games, the Stag Hunt coordination class, and the Prisoner’s Dilemma class, while only the Hawk-Dove and the Prisoner’s Dilemma are studied in Helbing and Yu (2008). We find that fully cooperative states can be reached for the standard neighborhoods and for several migration distances in the Stag Hunt case, while cooperation can also be achieved in the Prisoner’s Dilemma for a non-negligible part of its game space. Mobility is less beneficial in the hawk-dove game where cooperation levels are on the average only slightly better than in the static, motionless case. Finally, we also study the extreme case of system evolution when agents cannot change their initially attributed strategy and are only allowed to test free cells within their migration radius in order to possibly move to more profitable regions. Here cooperation cannot evolve by definition but we are interested in the dynamical patterns that may form, i.e. whether or not the agent distribution remains uniform during the dynamics. In this case, in the Prisoner’s dilemma and Stag Hunt games we find that players move in a coherent way because clusters of cooperators followed by defectors are formed. On the other hand, in the Hawk-Dove game or when the radius within which players move is small, players end up blocked into stationary clusters.

## Evolutionary Games and Migration in Two-Dimensional Space

### The Games Studied

We investigate three classical two-person, two-strategy, symmetric games classes, namely the Prisoner’s Dilemma (PD), the Hawk-Dove Game (HD), and the Stag Hunt (SH). These three games are simple metaphors for different kinds of dilemmas that arise when individual and social interests collide. The Harmony game (H) is included for completeness but it is not a dilemma since cooperation is trivially the NE. The main features of these games are summarized here

for completeness; more detailed accounts can be found elsewhere e.g. (Weibull, 1995; Hofbauer and Sigmund, 1998; Vega-Redondo, 2003). The games have the generic payoff matrix  $M$  (equation 1) which refers to the payoffs of the row player. The payoff matrix for the column player is simply the transpose  $M^T$  since the game is symmetric.

$$\begin{array}{cc} & \begin{array}{cc} C & D \end{array} \\ \begin{array}{c} C \\ D \end{array} & \begin{pmatrix} R & S \\ T & P \end{pmatrix} \end{array} \quad (1)$$

The set of strategies is  $\Lambda = \{C, D\}$ , where  $C$  stands for “cooperation” and  $D$  means “defection”. In the payoff matrix  $R$  stands for the *reward* the two players receive if they both cooperate,  $P$  is the *punishment* if they both defect, and  $T$  is the *temptation*, i.e. the payoff that a player receives if he defects while the other cooperates getting the *sucker’s payoff*  $S$ .

In order to study the usual standard parameter space (Santos et al., 2006; Roca et al., 2009), we restrict the payoff values in the following way:  $R = 1$ ,  $P = 0$ ,  $-1 \leq S \leq 1$ , and  $0 \leq T \leq 2$ .

For the PD, the payoff values are ordered such that  $T > R > P > S$ . Defection is always the best rational individual choice, so that  $(D, D)$  is the unique Nash Equilibrium (NE) and also the only fixed point of the replicator dynamics (Weibull, 1995; Hofbauer and Sigmund, 1998). Mutual cooperation would be socially preferable but  $C$  is strongly dominated by  $D$ .

In the HD game, the order of  $P$  and  $S$  is reversed, yielding  $T > R > S > P$ . Thus, when both players defect they each get the lowest payoff. Players have a strong incentive to play  $D$ , which is harmful for both parties if the outcome produced happens to be  $(D, D)$ .  $(C, D)$  and  $(D, C)$  are NE of the game in pure strategies. There is a third equilibrium in mixed strategies which is the only dynamically stable equilibrium (Weibull, 1995; Hofbauer and Sigmund, 1998).

In the SH game, the ordering is  $R > T > P > S$ , which means that mutual cooperation  $(C, C)$  is the best outcome and a NE. The second NE, where both players defect is less efficient but also less risky. The difficulty is represented by the fact that the socially preferable coordinated equilibrium  $(C, C)$  might be missed for “fear” that the other player will play  $D$  instead. The third mixed-strategy NE in the game is evolutionarily unstable (Weibull, 1995; Hofbauer and Sigmund, 1998).

Finally, in the H game  $R > S > T > P$  or  $R > T > S > P$ . In this case  $C$  strongly dominates  $D$  and the trivial unique NE is  $(C, C)$ . The game is non-conflictual by definition and does not cause any dilemma, it is mentioned to complete the quadrants of the parameter space.

There is an infinite number of games of each type since any positive affine transformation of the payoff matrix leaves the NE set invariant (Weibull, 1995). Here we study the customary standard parameter space (Santos et al., 2006; Roca



et al., 2009), by fixing the payoff values in the following way:  $R = 1$ ,  $P = 0$ ,  $-1 \leq S \leq 1$ , and  $0 \leq T \leq 2$ .

In the  $TS$ -plane each game class corresponds to a different quadrant depending on the above ordering of the payoffs as depicted in Fig 1, left image, and the figures that follow. The right part of Fig 1 shows the standard replicator dynamics results for a well mixed population (Weibull, 1995).

### Population Structure

The euclidean two-dimensional space is modeled by a discrete square lattice of side  $L$  with toroidal borders. Each vertex of the lattice can be occupied by one player or be empty. The *density* is  $\rho = N/L^2$ , where  $N \leq L^2$  is the number of players. Players can interact with  $k$  neighbors which lie at a distance smaller or equal than a given constant  $R_g$ . Players can also migrate to empty grid points at a distance smaller than  $R_m$ . The relationships between the neighborhoods defined as above and the customary square Moore neighborhoods of increasing order are illustrated in Fig. 2.

### Payoff Calculation and Strategy Update Rules

Here it is specified how individual's payoffs are computed and how agents decide to revise their current strategy. We take into account that each agent  $i$  interacts locally with a set of neighbors  $V_i$  lying closer than  $R_g$ . Let  $\sigma_i(t)$  be a vector giving the strategy profile at time  $t$  with  $C = (1, 0)$  and  $D = (0, 1)$  and let  $M$  be the payoff matrix of the game (equation 1). The quantity

$$\Pi_i(t) = \sum_{j \in V_i} \sigma_i(t) M \sigma_j^\top(t) \quad (2)$$

is the cumulated payoff collected by player  $i$  at time step  $t$ .

We use an asynchronous scheme for strategy update and migration, i.e. players are updated one by one by choosing a random player in each step with uniform probability and with replacement. Then the player migrates with probability  $p$  or updates its strategy with probability  $1 - p$ . Several update rules are customary in evolutionary game theory (Roca et al., 2009). Here we shall use imitative strategy update protocol which consists in switching to the strategy of the neighbor that has scored best in the last time step. This *imitation of the best* (IB) policy can be described in the following way: the strategy  $\sigma_i(t)$  of individual  $i$  at time step  $t$  will be

$$\sigma_i(t) = \sigma_j(t-1), \quad (3)$$

where

$$j \in \{V_i \cup i\} \text{ s.t. } \Pi_j = \max_{k \in \{V_i \cup i\}} \{\Pi_k(t-1)\}. \quad (4)$$

That is, individual  $i$  will adopt the strategy of the player with the highest payoff among its neighbors including itself. If there is a tie, the winner individual is chosen uniformly at random.

A final remark is in order here. The above model rules are common in numerical simulation work, which has the advantage that the mathematics is simpler and results can be compared with previous work. However, they are homogeneous among the agents and there is no learning. It is far from clear whether they are able to model real situations in biological systems and especially human societies. However, we feel that these considerations are outside the scope of the present numerical investigation.

### Strategy Imitation and Migration rules

When player  $i$  is chosen for update, she changes her strategy with probability  $1 - p$  or migrates with probability  $p$ . If the pseudo-random number drawn dictates that  $i$  should migrate, then she considers  $N_{test}$  randomly chosen positions in the disc of radius  $R_m$  around itself in order to take into account her bounded rationality.  $N_{test} = 20$  has been used in all the simulations. For each trial position the player computes the payoff that she would obtain in that place with her current strategy. The positions already occupied are just discarded from the possible choices. Then player  $i$  stays at her current position if she obtains there the highest payoff, or migrates to the most profitable position among those explored during the test phase. If several positions, including her current one, share the highest payoff she chooses one at random. The protocol described in Helbing and Yu (Helbing and Yu, 2009) is slightly different: the chosen player chooses the strategy of the best neighbor including himself with probability  $1 - r$ , and with probability  $r$  his strategy is randomly reset. Before this imitation step  $i$  deterministically chooses the highest payoff free position in a square neighborhood surrounding the current player and including himself. If several positions provide the same expected payoff, the one that is closer to the old position of  $i$  is selected.

---

#### Algorithm 1: migration of player $i$

---

```

for  $j \in [1, N_{test}]$  do
    choose random position  $x_j$  in  $V_i$ 
    if  $x_j$  is free then
        compute the expected payoff  $\Pi(x_j)$  of player  $i$ 
        at  $x_j$ 
    choose the best  $\Pi(x_j)$ ; if several  $x_j$  share the same
     $\Pi(x_j)$  choose one at random and migrate to this
    position
    
```

---

### Mobility Measure

In order to assess if a player has a definite direction of motion with respect to time we will use the following mobility measure. Mobility is defined as  $M = \max_{t \in [0, \tau]} (D_t) / L$  where  $\tau$  is the time interval for a player to travel a total distance  $L$  if she moves the maximal distance  $R_m$  at each time step in the same direction.  $D_t$  is the Euclidean distance from



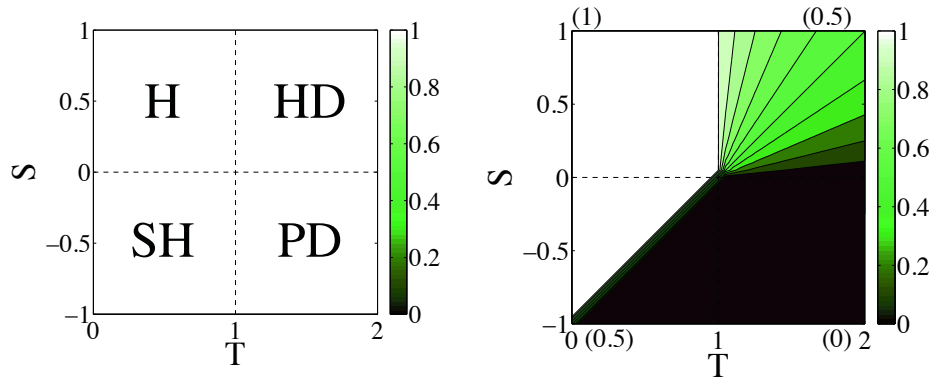


Figure 1: (Color online) Left image: The games phase space (H= Harmony, HD = Hawk-Dove, PD = Prisoner's Dilemma, and SH = Stag Hunt) as a function of  $S, T$  ( $R = 1, P = 0$ ). Right image: cooperation at steady state in a well mixed population for comparison purposes. Lighter tones stand for more cooperation. Figures in parentheses next to each quadrant indicate average cooperation in the corresponding game space.

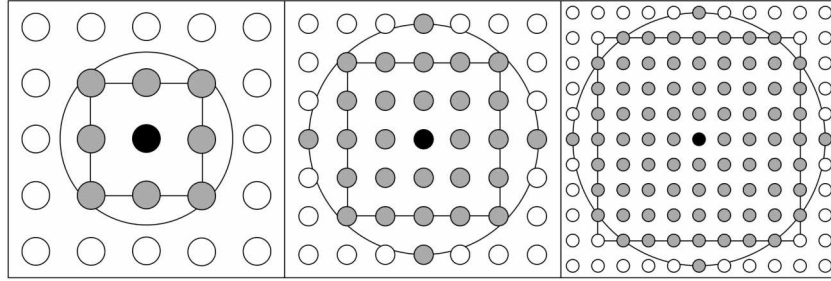


Figure 2: Relationships between the neighborhoods defined by the radii  $R_g$  and  $R_m$  and the Moore square neighborhoods. Left: with the 1.5 radius the neighborhood is identical with the standard Moore neighborhood at distance one. Middle: radius 3 is almost equivalent to a Moore neighborhood at distance two marked as a square. Right: with radius 5 the closer Moore neighborhood has distance four.

the initial position to the position at time  $t$ . The interval  $\tau$  is not taken from the beginning of the simulation but rather after a time sufficient for the mobile patterns to form. Thus  $M$  measures the ratio between the maximal distance over time reached by a player from her initial position and the maximal distance that it is possible to reach in the best case. We multiplied this measure by four in order to increase the contrast in the images. However, this measure is not a strict indicator of coherent motion as moving clusters can collide and change direction.

### Simulation Parameters

The  $TS$ -plane has been sampled with a grid step of 0.1 and each value in the phase space reported in the figures is the average of 50 independent runs. The evolution proceeds by first initializing the population by creating a player in each cell of the underlying lattice with probability  $\rho$ . Then the players' strategies are initialized uniformly at random such that each strategy has a fraction of approximately  $1/2$  unless otherwise stated. For each grid point, agents in the popula-

tion are chosen sequentially at random with replacement to revise their strategies or positions. Payoffs are constantly updated. To avoid transient states, we let the system evolve for a period of  $\tau = 1000$  time steps, for each time step  $N = 1000$  players are chosen for update. At this point almost always the system reaches a steady state in which the frequency of cooperators is stable except for small statistical fluctuations. We then let the system evolve for 50 further steps and take the average cooperation value, or the mobility, in this interval. We repeat the whole process 50 times for each grid point and, finally, we report the average cooperation values over those 50 repetitions.

## Results

### Strategy Evolution and Mobility

In this section we discuss cooperation results with the IB rule and adaptive migration and explore the influence of different radii  $R_m$  and  $R_g$  and the density  $\rho$ . Fig. 3 left image displays the cooperation level in the  $ST$ -planes with the

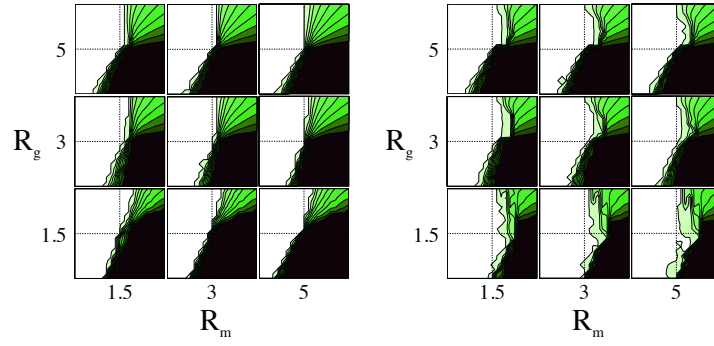


Figure 3: (Color online) Average cooperation levels with IB strategy revision rule as a function of  $R_g$  and  $R_m$  with  $p = 0.5$  and  $\rho = 0.5$ . Left image: Random migration. Right image: best fitness migration rule. The size of the population is 1000 players. In all cases the initial fraction of cooperators is 0.5 randomly distributed among the occupied grid points.

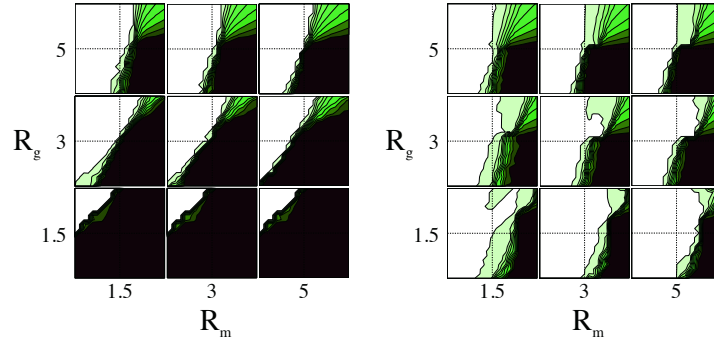


Figure 4: (Color online) Average cooperation levels with IB strategy revision rule as a function of  $R_g$  and  $R_m$  with  $p = 0.5$  and  $\rho = 0.1$ . Left image: Random migration. Second image: best fitness migration rule. The size of the population is 1000 players. In all cases the initial fraction of cooperators is 0.5 randomly distributed among the occupied grid points.

IB rule and a density  $\rho = 0.5$  for several combinations of  $R_g$  and  $R_m$ . For the sake of comparison, and in order to have a baseline case, Fig. 3 left image shows the case in which migration is not dictated by success but, rather, it is simply random, i.e. the target of migration will be a free cell randomly drawn among those contained in the  $R_m$  disk. The right image depicts cooperation levels when migration is success-driven. We see that, for  $R_g = 1.5$ , full cooperation is achieved in the SH quadrant for all  $R_m$  in the case of contingent migration while cooperation is notably lower in the random migration case for all  $R_m$ . For the PD cooperation remains nearly constant through  $R_m$  for  $R_g = 1.5$ , or slightly improves with smaller  $R_m$  in the contingent migration case with average values in the quadrant of about 0.3. In contrast, it is almost zero in the random diffusion case. Increasing the game radius  $R_g$  doesn't help and all average values tend to fall independent of  $R_m$ . This is because enlarging the neighborhood of a player is a step towards the mixed population in which cooperation results are worse, as can be seen in Fig. 1. We have observed that the increase in cooperation for  $R_g = 1.5$  with “intelligent” migra-

tion is essentially due to the formation of cooperator clusters that remain relatively stable throughout evolution thanks to the possibility for cooperators to join one of those clusters. With larger  $R_g$  values, small cooperator clusters are easier to break and large  $C$  clusters, which would help cooperation to establish itself in the cluster, cannot form and defection prevails at least in the PD case. The Hawk-Dove game, due to its mixed-strategy equilibrium benefits less from success-driven migration as the two other games.

Density is a parameter that heavily influences the evolution of cooperation Vainstein et al. (2007); Sicardi et al. (2009), also in the presence of intelligent migration Helbing and Yu (2009); Jiang et al. (2010). Too high densities are detrimental because they tend to limit the mobility of agents to a point that only cooperator clusters that appear owing to statistical fluctuations in the initial population compositions can eventually remain stable. It appears that low and intermediate densities give more freedom to the population for moving around and to search for better positions. Figure 4 right image shows average cooperation results for the IB strategy revision rule and the same combinations of radii

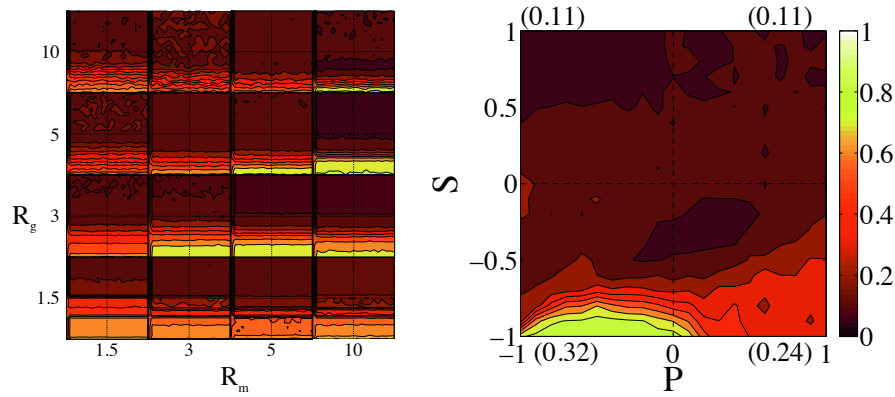


Figure 5: (Color online) Left: average mobility levels in the ST-plane as a function of  $R_g$  and  $R_m$ . Right: mobility in the SP-plane for  $T = 1$ ,  $R_m = 10$  and  $R_g = 10$ . The best fitness migration rule is used. The size of the population is 1000 players and  $\rho = 0.1$ . Cooperators fraction is 0.5 randomly distributed among the population. Lighter tones stand for more mobility.

but for  $\rho = 0.1$  instead of 0.5. With  $\rho = 0.1$  cooperation generally increases. In this case defectors attack clusters with a smaller rate since they are more diffused in space and move randomly until they find a cluster of cooperators. The advantage of intelligent migration with respect to random motion is even more marked here by comparing with the left image. In the latter defection appears to be even stronger than in the mixed population, but this is rather special and is due to the fact that the system is very diluted which causes most encounters to be between just two players.

### Mobility Only: Emergence of Dynamic Clusters

In this section we study the emergence of dynamic clusters. These clusters are formed by a cohesive group of cooperators followed by defectors. The left image of Fig. 5 displays the mobility of nodes (see Sect. 5) for several ST-planes as a function of the game and migration radius. Lighter tones stand for higher mobility and indicate that such dynamic clusters may form. It can be observed that the dynamic clusters tend to appear with low  $S$ . The horizontal stripes of constant  $M$  can be explained by the fact that, as long as  $P = 0$ , all positive values of  $T$  are identical since the best target position for migration remains the same. On the other hand, when  $P$  is comparable to  $T$  or larger, defectors form clusters among themselves and stop following cooperators, which causes  $M$  to decrease. In contrast, when  $P$  is negative enough, defectors repel each other and they can not gather behind cooperators. These effects are reflected in the averages shown in the right image of Fig. 5.

We display dynamic clusters for some particular runs in Figs. 6 and 7. Figure 6 shows clusters that have formed after a number of time steps and that are already stable as a function of  $S$  with  $R_g = 5$  and  $R_m = 10$ . The corresponding game can be inferred from Fig. 5 left image. From left to right the images show situations with increasing cluster mobility. There is a sort of mobility transition such that,

while the first two images show clusters that do not move, the rightmost one corresponds to a situation in which the clusters are much more dynamical.

Figure 7 shows the clusters appearance when mobility is high (compare with Fig. 5 left image) as a function of the radius of play  $R_g$  for the same game as above, which is in the PD region. One can see that there is a direct relationship between increasing  $R_g$  and the cluster size. With a given  $R_m$ , which is here 10, when  $R_g$  is comparatively small, clusters do form but they are continuously destroyed and reformed in an other places without a definite motion.

**The Effect of Strategy Update** In the limiting case  $p \rightarrow 1$ , i.e. very little strategy update with respect to migration, dynamical patterns form before any significant strategy update. Fig. 8 displays the ST-plane in that case. It can be observed that cooperation is lost for the lower values of  $S$ . This loss of cooperation can be related to an increase in mobility by comparing Fig. 4 (right) with Fig. 8 and by remarking that the loss of cooperation between these two cases correspond to the relatively high levels of mobility seen in Fig. 5 for this area of the game space. In the case  $p = 0.5$  the dynamical patterns cannot fully form since the strategy evolution is too fast. In fact as clusters of cooperators form defectors are attracted towards them. Considering only the case in which cooperation thrives, if  $p$  is high enough the incoming defectors are transformed into cooperators directly while approaching the cluster. Thus the cluster remains static and grows. On the other hand, when  $p$  is small the migrating defectors cumulated around the cluster will eventually cause it to move.

### Conclusion

In the framework of game theory we have studied the evolution of cooperation in spatially structured populations when a given focal player can only interact with players contained

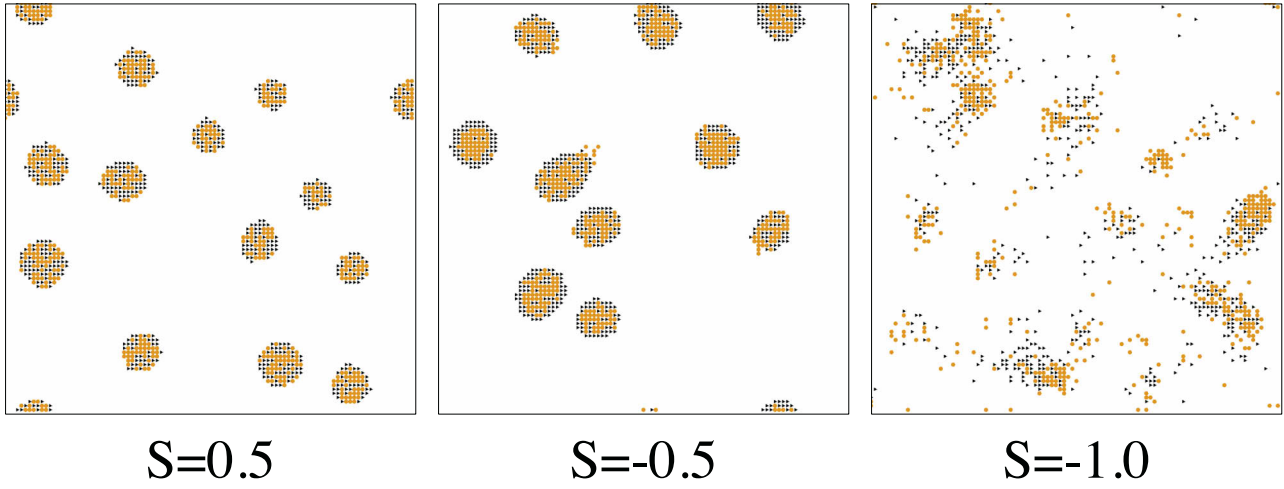


Figure 6: (Color online) Cluster for  $R_g = 5$  and  $R_m = 10$ ,  $T = 1.5$ . Cooperators are represented as orange circles and defectors as black triangles.

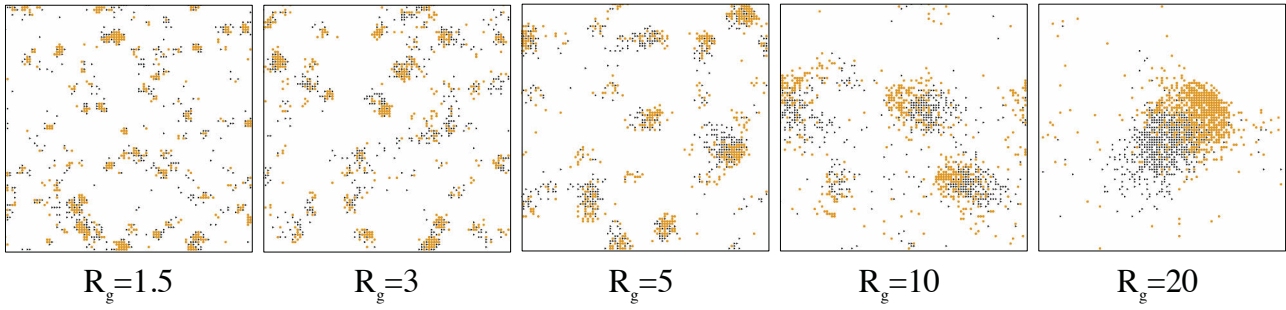


Figure 7: (Color online) Cluster for  $R_m = 10$ ,  $T = 1.5$ ,  $S = -1.0$  for different  $R_g$  values. Cooperators are represented in orange and defectors in black.

in a radius  $R_g$  centered on the focal player that is small with respect to the space available. This locality of interactions is a realistic feature of actual populations and markedly differs from the customary well mixed population. Besides being able to adapt their strategy with probability  $p$ , in our model players can also move around to unoccupied places in the underlying two-dimensional grid also with probability  $1 - p$ . The amount of displacement is determined by the migration radius  $R_m$ . Migration depends on the payoff, i.e. a player that has decided to migrate can examine a number of free positions around it within the radius  $R_m$ , earn a potential payoff by fictitious play with the neighbors at that position, and finally choose to migrate to the position that provides the best payoff among those tested. We show that an equal amount of this strategy and of strategy mutation in the original position gives rise to full cooperation in the SH game

space and, to comparatively high values in the more difficult PD game space. This is particularly striking when compared with the baseline case in which strategy revision is identical but migration is to a randomly chosen free cell in the disk of radius  $R_m$ .

We have also investigated pattern formation in the population under the effects of intelligent migration only. In this case too we start from a 50 – 50 random distribution of cooperators and defectors. However, now cooperation cannot evolve since strategy changes are not allowed. What we do observe is a very interesting and intricate phenomenon of dynamical or almost static pattern formation that is related to the underlying game played and that also depends on the  $R_g$  and  $R_m$  radii. We have analyzed the nature and dynamics of these clusters and we have shown that mobility of agents can be high when the sucker payoffs  $S$  reaches negative enough

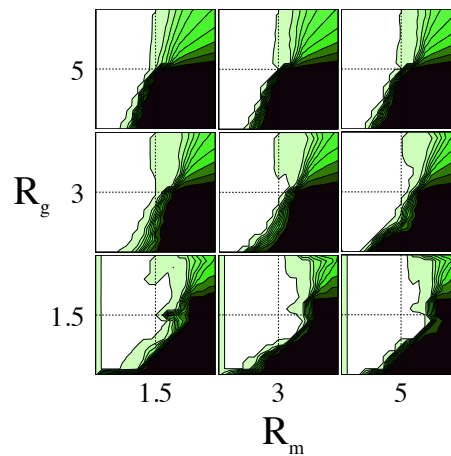


Figure 8: (Color online) Average cooperation levels with best fitness migration rule and IB strategy update rule as a function of  $R_g$  and  $R_m$ ,  $\rho = 0.1$ , and  $p = 0.99$ . The size of the population is 1000 players. The initial fraction of cooperators is 0.5 randomly distributed among the occupied grid points.

values compared to the reward payoff  $R$ . The temptation  $T$  has only to be positive as the punishment  $P$  is null in our settings. For high interaction radius  $R_g$  and migration radius  $R_m$  the motion is coherent and the cooperators tend to gather and move in the same direction with swarms of defectors following them. When  $R_m$  is small players can be blocked into clusters. On the other hand, when  $R_g$  is low and  $R_m$  high the clusters are constantly destroyed and reformed in different places. For both  $R_g$  and  $R_m$  low small clusters are formed and the motion is not definite. Future work should include the study of the effect of strategy update and mobility noise in the dynamics, as well as the use of different strategy update and, possibly, migration rules.

### Acknowledgments

The authors wish to thank the Swiss National Foundation for their financial support under contract n. 200021-14661611.

### References

- Aktipis, C. A. (2004). Know when to walk away: contingent movement and the evolution of cooperation. *J. Theor. Biol.*, 231:249–2160.
- Axelrod, R. (1984). *The Evolution of Cooperation*. Basic Books, Inc., New-York.
- Buesser, P. and Tomassini, M. (2012). Evolution of cooperation on spatially embedded networks. *Phys. Rev. E*, 86:066107.
- Chen, Z., Gao, J., Kai, Y., and Xu, X. (2011). Evolution of cooperation among mobile agents. *Physica A*, 390:1615–1622.
- Enquist, M. and Leimar, O. (1993). The evolution of cooperation in mobile organisms. *Anim. Behav.*, 45:747–757.
- Helbing, D. and Yu, W. (2008). Migration as a mechanism to promote cooperation. *Advances in Complex Systems*, 11:641–652.
- Helbing, D. and Yu, W. (2009). The outbreak of cooperation among success-driven individuals under noisy conditions. *Proc. Natl. Acad. Sci. USA*, 106:3680–3685.
- Hofbauer, J. and Sigmund, K. (1998). *Evolutionary Games and Population Dynamics*. Cambridge, N. Y.
- Jiang, L.-L., Wang, W.-X., Lai, Y.-C., and Wang, B.-H. (2010). Role of adaptive migration in promoting cooperation in spatial games. *Physical Review E*, 81:036108.
- Nowak, M. A. and May, R. M. (1992). Evolutionary games and spatial chaos. *Nature*, 359:826–829.
- Nowak, M. A. and Sigmund, K. (2000). Games on grids. In Dieckmann, U., Law, R., and Metz, J. A. J., editors, *The Geometry of Ecological Interactions: Simplifying Spatial Complexity*, pages 135–150. Cambridge University Press, Cambridge, UK.
- Roca, C. P., Cuesta, J. A., and Sánchez, A. (2009). Evolutionary game theory: temporal and spatial effects beyond replicator dynamics. *Physics of Life Reviews*, 6:208–249.
- Santos, F. C., Pacheco, J. M., and Lenaerts, T. (2006). Evolutionary dynamics of social dilemmas in structured heterogeneous populations. *Proc. Natl. Acad. Sci. USA*, 103:3490–3494.
- Sicardi, E. A., Fort, H., Vainstein, M. H., and Arenzon, J. J. (2009). Random mobility and spatial structure often enhance cooperation. *J. Theor. Biol.*, 256:240–246.
- Vainstein, M. H., Silva, A. T. C., and Arenzon, J. J. (2007). Does mobility decrease cooperation? *J. Theor. Biol.*, 244:722–728.
- Vega-Redondo, F. (2003). *Economics and the Theory of Games*. Cambridge University Press, Cambridge, UK.
- Weibull, J. W. (1995). *Evolutionary Game Theory*. MIT Press, Boston, MA.



## *In silico* evolution of transferable genetic elements

Dusan Misevic<sup>1\*</sup>, Antoine Frénay<sup>1</sup> and François Taddei<sup>1</sup>

<sup>1</sup>Center for Research and Interdisciplinarity, INSERM U1001, University Paris Descartes, Sorbonne Paris Cité, France

\*corresponding author: dule@alife.org

### Abstract

Plasmids are an integral and essential factor in microbial biology and evolution, with broad implications ranging from antibiotic resistance to research tools. Much has been done to describe, quantify, and modify properties of transferable plasmids, including the extensive theoretical work using simulations and models. However, a wide gap between theory and experiments still remains, especially relating to the underlying genetic architecture of transfer as well as coevolutionary dynamics of the plasmid infectivity and susceptibility. Large-scale genomic studies and more biologically accurate models are among different approaches working towards narrowing this gap. Here we describe how *Aevol*, a digital evolution system, can be effectively used to study plasmid and quantify various aspects of their evolution and its outcomes. Specifically, we find that plasmid maintenance is extremely sensitive to the direct fitness cost of expressing transfer genes. In our study, the genes for donor ability and recipient immunity (which additively describe the probability of plasmid transfer) typically, but not exclusively, evolved on the plasmid itself. Additionally, we find epistatic interactions between genes on plasmids and the chromosome may evolve, a new aspect of their interaction and struggle for control over each other. There is a strong coevolutionary link between donor ability and recipient immunity, with their values tracking and being driven by one another. While plasmids seem to largely behave as selfish genetic elements, they occasionally may also carry metabolic genes and directly increase individual's fitness. With a number of concise questions and results, this initial study of plasmids in *Aevol* establishes the baseline and opens possibilities for future work, while simultaneously uncovering and describing novel evolutionary trajectories taken by the transferable genetic elements.

### Introduction and Background

Horizontal gene transfer in general, and plasmid conjugation in particular, have been identified as major mechanisms in microbial evolution (Ochman et al. 2000; Koonin and Wolf 2012). Better understanding of the movement of the genetic material between different species has fundamentally changed how we view and analyze phylogeny of life (Doolittle 1999; Koonin et al. 2001; Ragan et al. 2009). In parallel, plasmids have been the focus of extensive research due to their role in acquisition, maintenance, and transfer of antibiotic resistance (Davison 1999; Alekshun and Levy 2007; Bennett 2008). They have also been harnessed as a powerful tool in molecular biology, enabling research ranging from creation of

synthetic genes and gene therapy to design of pet glow-in-the-dark fish (Cohen et al. 1973; Sambrook et al. 1989; Pray 2008; Constante et al. 2011). From suicide plasmids to plasmid addiction, these transferable bits of DNA hold seemingly inexhaustible diversity of strategies for spread and survival, making them an intriguing and fascinating subject of research (Lipps 2009).

Given their spread and importance in the natural world, it is no great surprise that plasmid biology has been extensively modeled over the past decades, primarily with analytical models using differential equations (Stewart and Levin 1977; Levin and Stewart 1980; Bergstrom et al. 2000). One of the main limitations of such models rests in their lack of spatial structure – organisms interact with each other at random and cannot preferentially associate with each other. More recent work has addressed these issues by simulating the spatial dynamics of plasmids on lattices (Krone et al. 2007) and cellular automaton-like graphs (Connelly et al. 2011). However, in all of these studies, the individuals were just a collection of numerical parameters such as their plasmid susceptibility, birth or death rate, but did not have a genome and were thus not well suitable for understanding potentially important consequences of the genetic architecture of plasmid conjugation. Previous models were unable to consider the location of the plasmid transfer genes, how they may move between the plasmid and the chromosome, or interact depending on their location, which we remedy here.

Classically, plasmids can be placed into two distinct categories, based on their ability to transfer themselves from one individual to another: the conjugative plasmids, which carry the genes enabling the transfer themselves, and mobilizable plasmids, which require other means, such as genes located on other plasmids. Rather than focus on one or the other type, in our research system we give the transfer genes the opportunity to evolve on either the main chromosome or the plasmid, as well as to be freely exchanged between the two. Additionally, our digital plasmids can carry both metabolic and transfer genes and thus effectively control their horizontal and vertical transmission by modifying infectivity or directly changing the host's fitness advantage in the population. In a series of experiments presented here, we evolve and analyze hundreds of populations for thousands of generations and are able to characterize the diversity of evolutionary strategies for the location and effect of genes controlling the *in silico* plasmid transfer.

## Methods

To study plasmid dynamics and evolution we use *Aevol*, digital experimental platform that enables us to maintain, track, and manipulate large populations of digital organisms over thousands of generations. *Aevol* is similar to and builds on the success of the existing digital experimental systems, such as Avida (Lenski et al. 1999; Misevic et al. 2004, 2006; Goldsby et al. 2012). However, it also includes significant changes, particularly pertaining to more biologically realistic genetic encoding and genotype-phenotype mapping (Knibbe et al. 2006; Knibbe et al. 2007; Knibbe et al. 2008; Beslon et al. 2010). *Aevol* is freely available for download at [www.aevol.fr](http://www.aevol.fr). In all our experiments we used the default parameters unless otherwise noted. Main properties of *Aevol* have been described in great detail previously (Parsons et al. 2010; Misevic et al. 2012), so here we focus on the features specifically implemented and directly relevant for our study of the plasmids evolutionary dynamics.

### The Aevol experimental system

#### General properties.

*Aevol* individuals are double stranded binary strings, typically thousands of base pairs long. An evolved organism contains multiple proteins, flanked by promoters, terminators, and start/stop codons. During its lifetime, digital genomes undergo a microbial genetic inspired transcription and translation steps to determine the their phenotype and in turn organisms' fitness. The phenotype is the collection of proteins, each an abstract entity, physically represented by a triangle located on the phenotypic axis. The phenotypic axis is the collection of all possible traits that may be a part of the organism's phenotype, each trait corresponding to a real number between 0 and 1. Two traits that are next to each other on the phenotypic axis are more likely to interact pleiotropically, as there is a higher chance that a single protein would affect them both, than two traits located further away. Two different sequences may encode for the same gene but the traits that are close to each other on the phenotypic axis are not necessarily encoded by genes with high sequence similarity. Each protein triangle has three properties: (1) the location on the phenotypic axis that signifies the trait it primarily affects, (2) the height that represents its expression level, and (3) the width that shows the range of neighboring traits it also affects.

For any specific experimental environment, there exists a single, constant target phenotype, which is a collection of trait expression levels that are most optimal for this environment and are the target for selection. The fitness of an individual is calculated as the difference between the target function and the phenotype, and intuitively represents the percentage of the area under the target phenotype function that is covered by the phenotype. The fitness can theoretically be negative, for example for individuals that express proteins with optimal levels of zero, but such individuals are rare and are quickly selected out of the population.

**Population structure:** The default *Aevol* populations do not have an explicit structure and are akin to well-mixed bacterial populations living in liquid media. However, as spatial structure is thought to play an important role in plasmid transfer (Krone et al. 2007), we are using the square grid

structure originally implemented for the study of cooperation (Misevic et al. 2012). We can vary the strength of spatial structure using a migration parameter (*mig*), which determines the number of swaps that happen at every generation. For each swap we choose at random a pair of organisms in the population and exchange their location. High *mig* (on the order of population size) thus corresponds to a well-mixed population, while a low one (*mig* = 0) a perfectly spatially structured one.

**Mutations, selection, and reproduction:** The genome of a new organism may be different from its parent due to mutations, the errors made during replication. Specifically, organisms experience small mutations (point mutations and insertion/deletions of sequences up to 6 base pairs) as well as large mutations (duplications/deletions of more than 6bp, translocations and inversions). *Aevol* is a synchronous evolutionary model, so the fitness of all individuals is evaluated at the same time, just prior to selection. Each organism competes with neighbors from the classical 3x3 Moore neighborhood around it for a chance to place its offspring in the next generation. The offspring is chosen using roulette selection on the probabilities derived from fitness of all the individuals in the neighborhood. Each individual has a  $(a - 1) \times a^{9-R} / (a^9 - 1)$  probability of reproducing into the central position of the neighborhood, where *R* is the organism's fitness-based rank and *a* is the population-level selection pressure constant. We should note that while the phenotypic target is fixed during each experiment, the effective strength of selection does not necessarily decrease or plateau. As target expression levels are real numbers and the selection we use is rank-based, even the smallest differences in fitness will be selected for, leading to continual adaptation, if not a truly open-ended evolution. Following selection, all individuals are reproduced simultaneously, with mutations, and placed into the population.

#### Plasmids in Aevol

The most fundamental property of *Aevol* plasmids is that they are treated as genetic units, equivalent to the already existing chromosome in all but one aspects of their digital biology. They mutate at the same rate as the chromosome, are transferred vertically to the offspring during reproduction, and the genes encoded on the plasmid are combined with the chromosomal genes to form the organism's phenotype. For large mutations, for example transposition, the beginning and the end of the transposed segment, as well as the location where it will be inserted, are chosen at random from the combined length of the chromosome and plasmid. As only the beginning and the end, but not the insertion location of the transposon, must be on the same genetic unit, this mutational mechanism allows for genes to freely move between the genetic units.

The exception from the equality between the chromosome and the plasmid is, of course, that plasmids are mobile genetic elements and may also transfer horizontally, between neighboring individuals in the same generation, while chromosomes cannot do so. Both plasmids and chromosomes are capable of controlling the rate of this transfer, inspired by bacterial conjugation, by evolving genes to decrease or increase the probability of sending or receiving a plasmid.

*Donor ability and recipient immunity:* In order for individuals to be able to control the rate of plasmid conjugation, we split the phenotypic axis into three sections: metabolism, donating, and receiving. The proteins located on the metabolism section of the axis directly affect the fitness, based on how closely they match the target phenotype, as they do on the single-section axis in classical *Aevol* experiments. The proteins on the donating or receiving sections determine the organism's donor and recipient ability. The donor ability corresponds to the effort that the organism will exert in order to transfer the plasmid. Inversely, the recipient ability, which we will refer to as plasmid exclusion ability or immunity, corresponds to the effort that an individual will put into not accepting the plasmid being sent to it. Both are calculated analogously to the metabolic fitness and represent the percentage of area under the target curve in the appropriate axis section that is covered by the protein triangles. Ancestral organism has no proteins associated with plasmid transfer, but they can appear via random mutations, spread due to positive, or be eliminated via negative selection.

*Plasmid conjugation:* Plasmid transfer in *Aevol* may happen only between individuals that share the same 3x3 Moore neighborhood. At every generation, all individuals have a chance to transfer their plasmid. A given, focal organism, is first queried for its donor ability. The probability of transfer to each of the neighboring individuals is equal to the difference between focal individual's donor ability and the chosen neighboring individual's recipient immunity. If the recipient immunity is greater than the plasmid sender's donor ability, the transfer will not happen. When plasmid is transferred, a copy of the plasmid from the donor is made and it replaces the plasmid that is located in the recipient. As with other population level processes in *Aevol*, all individuals attempt to transfer their plasmids simultaneously. However, as the conjugation algorithm is necessarily executed sequentially, we must avoid giving a higher chance to individuals and plasmids that transfer first then to ones whose plasmids may have already been replaced by the time they try to transfer them. To do so, we randomize the order at which individuals attempt to transfer their plasmid at every generation. Finally, in nature there are examples of retrotransfer, where plasmid recipient also transfers genetic material back to the donor (Sia et al. 1996; Szpirer et al. 1999). Inspired by this, we included a parameter that specifies whether populations evolve with unidirectional (plasmid replacement) or bidirectional (plasmid swap) conjugation mechanism.

In the majority of our experiments, the organisms did not pay any direct cost for expressing the genes for plasmid transfer, the same way they do not pay any explicit cost for expressing metabolic genes. However, in nature, pili and other conjugation-related machinery is not only costly to produce but may also have a detrimental effect as it serves as a target for phage attachment (Smillie et al. 2010), so we added a fitness cost for expression of the donor/recipient genes, proportional to the donor/recipient ability they confer. We should stress that this is not a cost of transfer, as it affects the fitness no matter whether the plasmid transfer successfully happens or not.

*Plasmid copy number and loss:* For the ease of implementation, data collection and analysis, we assume that all individuals in *Aevol* have a single plasmid. Examining the

effects of plasmid copy number would certainly be interesting, but remains as a potential topic of future studies. Similarly, we chose not to directly model plasmid loss, but individuals still may effectively lose their plasmid by drastically decreasing its size. The smallest gene in *Aevol* requires at least 48 base-pairs, which includes 22bp-long promoter, 6bp Shine-Dalgarno sequence, 4bp Shine-Dalgarno spacer, start codon, three codons determining the width, height, and mean of the protein triangle, stop codon (all codons are 3bp long), and 11bp reverse-complement terminator sequence. Thus, while technically plasmids cannot be entirely lost in *Aevol*, the organisms can evolve to not use and effectively eliminate them.

We conclude the description of *Aevol* and its the genetic algorithm heuristic by recounting the events that occur during a single generation of evolution: (1) organisms' fitness is evaluated, based on their metabolic proteins, (2) organisms that will reproduce are selected based on their fitness, (3) mutations are applied to the new-born organisms, (4) organisms migrate, by exchanging places with a randomly chosen individual, (5) plasmids are transferred between pairs of neighboring individuals, based on their donor ability and recipient immunity, followed by the start of the next generation and return to first event in the cycle. Using this setup throughout our experiments we are able to study the dynamics of plasmid conjugation in *Aevol* over thousands of generations of evolution.

### Experimental design

Given the number of parameters relevant for plasmid transfer that can potentially be varied in *Aevol*, it is computationally nonpermissive to examine all possible combinations in any type of a factorial experimental design. Instead, we first focus on the interaction of plasmid conjugation and population structure. To do so, we performed experiments in which we set the rate of migration to 0, 100, 300, or 1000. In these experiments, the organisms did not pay any direct cost of expressing the donating or receiving genes. In the second set of experiments, we had no migration, but varied the cost of transfer instead, from 0 to 0.03, 0.1, or 0.3. To evaluate the interaction between the donor and recipient abilities, and potential coevolutionary dynamics, we also ran experiments in which either donor or recipient ability was not allowed to evolve. In particular, we set the extrinsic, constant probability of plasmid donation to 1, 0.3, 0.1, or 0.01 for all individuals, but made any genes on the donor part of the phenotypic axis act as neutral and not have any effect on fitness or donor ability. Alternatively, we set the default probability of transfer to 0 and made recipient genes act as neutral ones, allowing only for the evolution of donor ability. Finally, we conducted experiments in which the plasmid transfer was not unidirectional and instead of the invading plasmid replacing the resident one, the plasmids from the two individuals swapped places.

For each set of parameters we performed 20 replicate experiments by evolving populations of 1024 individuals for 20,000 generations. The replicate populations were started with a randomly generated ancestor containing a single metabolic gene. Each population was associated with a different seed for the random number generator, which governs all stochastic processes during evolution. All

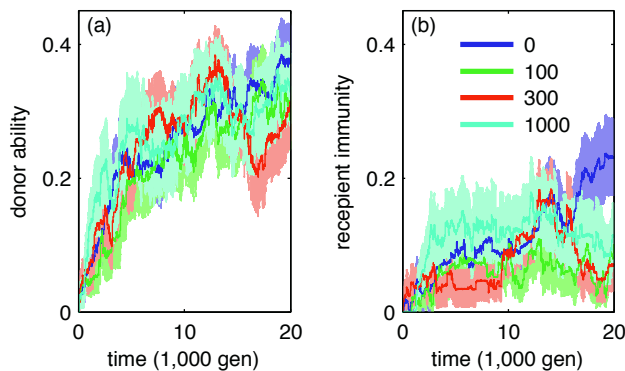
populations shared the same phenotypic target function, specified by the arithmetic sum of six Gaussian functions of the form  $y = H \exp(-(x - M)^2 / (2W^2))$ , where  $(H, M, W) = \{(0.25, 0.15, 0.04), (0.35, 0.2, 0.02), (0.35, 0.45, 0.02), (0.25, 0.5, 0.04), (0.35, 0.8, 0.02), (0.25, 0.85, 0.04)\}$ . First pair of functions is located on the metabolism part of the phenotypic axis, while the second and third pairs are on the donating and receiving sections, respectively. The three sections of the axis are equal in length. The mutation rate per base-pair was  $2.5 \times 10^{-5}$  for all small and  $2.5 \times 10^{-6}$  for all large mutations. Selection pressure constant was  $\alpha = 0.7$ ;

At each generation, we recorded the average values for fitness, genome length, donor ability and recipient immunity in the population. Additionally, we recorded each of these values not only for the entire organism, but also for individual genetic units separately. For example, for an individual we would calculate the donor ability it would have if it contained only the plasmid or only the chromosome. All the statistical analysis was performed using Matlab R2012b.

## Results and discussion

### Plasmids and migration

In the first set of experiments we studied the effect of population structure on the evolution of plasmid transfer. Our expectation was that higher donor ability, and thus higher rate of transfer, would evolve in populations with no migration. Specifically, we thought that in the competition between donor ability and recipient immunity genes, the former would be favored in spatially structured populations by clustering together individuals that transfer genes to each other and thus decreasing the probability of negative interaction between imported and resident genes. The data in part did not support

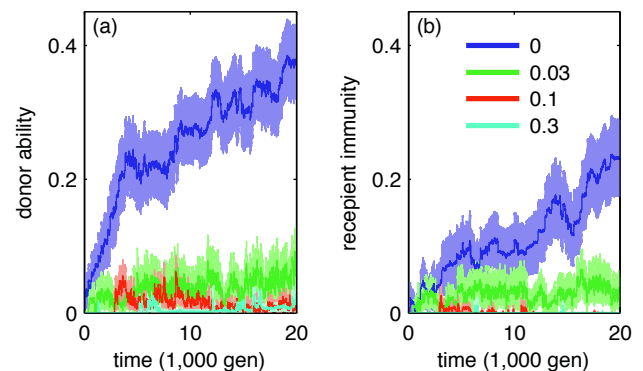


**Figure 1. Effect of population structure on donor ability and recipient immunity of plasmid transfer.** Each line is an average value for (a) donor ability and (b) recipient immunity across 20 replicate populations that evolved for 20,000 generations with the same set of parameters, but a different random number seed. Different colors represent different migration rates (the number of organisms pairs that get swapped at each generation, 0, 100, 300, or 1000). The shaded area around the lines represents one standard error of the mean. The no migration treatment (dark blue) was used in all future sets of experiments as the baseline.

our intuition (Figure 1): the average donor ability was not statistically different between the treatments at the end of the experiments (two-sample t-test,  $p > 0.2$  for all pairwise comparisons between treatments), and in all but one case, neither was the recipient immunity (two-sample t-test,  $p > 0.05$  except for the comparison between  $mig = 0$  and  $mig = 300$ , where  $p = 0.047$ ). However, the average probability of transfer did generally increase with higher migration (mean value of 0.144, 0.191, 0.243, 0.220 for  $mig = 0, 100, 300, 1000$ , respectively). Overall, in spite of much variation between the replicate experiments, we do find a trend for migration positively affecting the probability of plasmid transfer (all pairwise comparisons are significant, two-sample t-test, with at least  $p < 0.03$ ). Additionally, these differences did not come solely from the change in donor ability or recipient immunity, but the interaction between the two. As transfer happens within a generation and organisms' migration happens between generations in *Aevol*, we cannot account for physical processes that could impede conjugation, such as pili breakage or detachment. Instead, our work suggests transfer is more favored when it can also result in the transferred genes spreading further and faster due to organism migration. These and other selection forces that may increase conjugation in well-mixed populations should be investigated in greater detail.

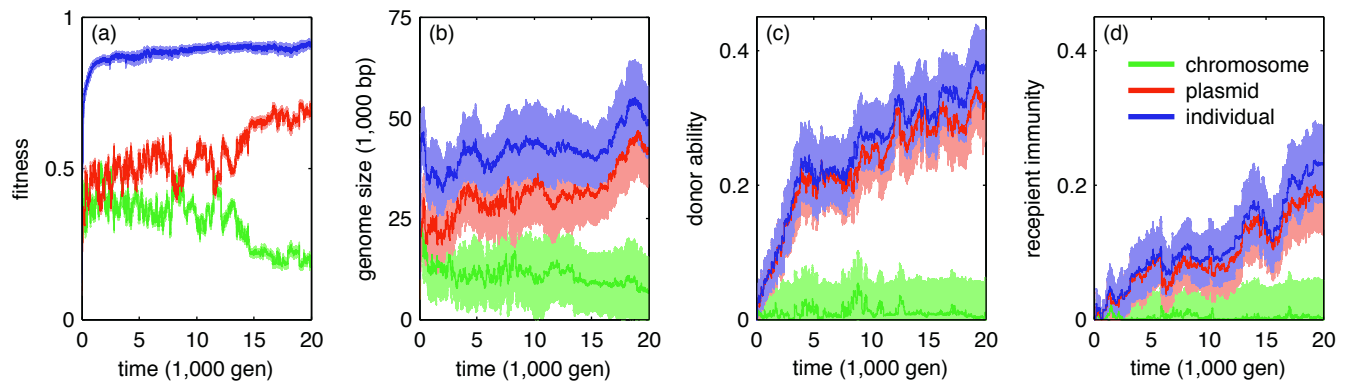
### Cost of transfer

We continued to quantify the properties of *in silico* transfer by varying the cost of expressing the genes that are involved in moderating the transfer rate. For each transfer-related gene the fitness of a digital individual would decrease proportionally to the product of cost and the increase of the donor ability or recipient immunity. Our expectations were clear: at higher costs of expressing donor genes, less plasmid transfer would evolve. Data supports our hypothesis (Figure 2) and we find



**Figure 2. Effect of cost on donor ability and recipient immunity of plasmid transfer.** Each line is an average value for (a) donor ability and (b) recipient immunity across 20 replicate populations that evolved for 20,000 generations with the same set of parameters. Different colors represent costs of expressing donor ability or recipient immunity genes (0, 0.03, 0.1, or 0.3). The shaded area around the lines represents one standard error of the mean. The data for the baseline populations (dark blue lines, no cost) are the same as the ones used in the Figure 1 (no migration, dark blue line).





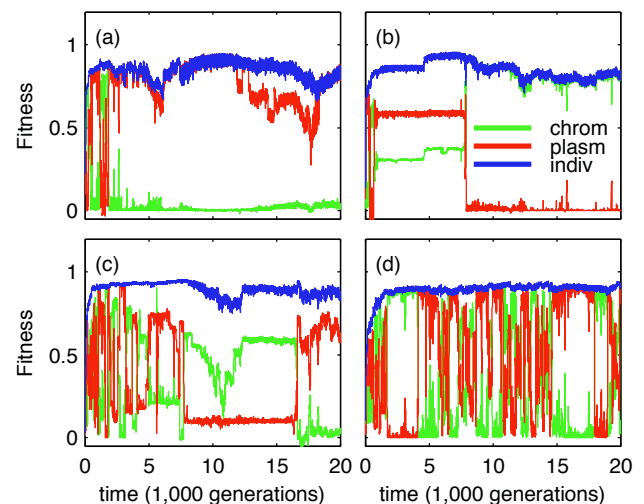
**Figure 3. Genetic architecture of transfer.** Four basic population properties, (a) fitness, (b) genome size, (c) donor ability, and (d) recipient immunity are averaged across 20 replicate populations that evolved for 20,000 generations with the same set of parameters. Different colors represent the property values obtained when considering just the chromosome (green), just the plasmid (red), or the entire organism (blue). The shaded area around the lines represents one standard error of the mean. The data for the baseline population (dark blue lines) are the same as the ones used in the baseline runs (no cost, no migration) in the previous figures.

that most transfer evolves when cost was zero ( $p < 0.001$  for comparison with all other treatments), while low levels of transfer evolve at cost 0.03, and effectively no transfer at higher costs. While orders of magnitude less than the rates of transfer in nature, given our relatively small population sizes, the evolved number of transfers per generation is comparable to natural populations. We should note that even when explicit cost of transfer genes was zero, genes that do not increase fitness are quickly lost in *Aevol*, both due to drift and because they may interfere with other genes through large mutations (Knibbe et al. 2007). Given our results and the wide spread of transferable plasmids in nature, we could speculate that their cost of expression may not be greatly different than one incurred by any other genes. Alternatively, such cost could be offset by some direct and strong benefit that plasmids would confer, such as the frequently observed antibiotic resistance (Svara and Rankin 2011).

### Genetic architecture of transfer

We continue the analysis of the baseline runs, with no migration or transfer cost, by examining the location of the metabolic and transfer genes. In nature, the molecular machinery for plasmid conjugation is located on the plasmid itself (Zatyka and Thomas 1998), but individuals could also control plasmid transfer via genes located on the chromosome. We expect that if plasmids confer a cost on their hosts, genes for recipient immunity would be selected for, and located on the main chromosome. In our baseline runs (where individuals evolved without any migration or transfer cost), we observe that the plasmid is the dominant genetic unit of the individual: not only are transfer genes located on the plasmid (Figure 3c, 3d), but it is on average larger than the chromosome (Figure 3b) and carries the majority of metabolic genes (Figure 3a). All the differences are statistically significant, as determined by two-sample t-test, with  $p < 0.01$ . We conclude that plasmids in *Aevol* behave largely like selfish genetic elements, controlling and increasing their own spread, but also taking on other aspects of the organisms' phenotype. Rather than the chromosome being the one who is trying to exclude invading

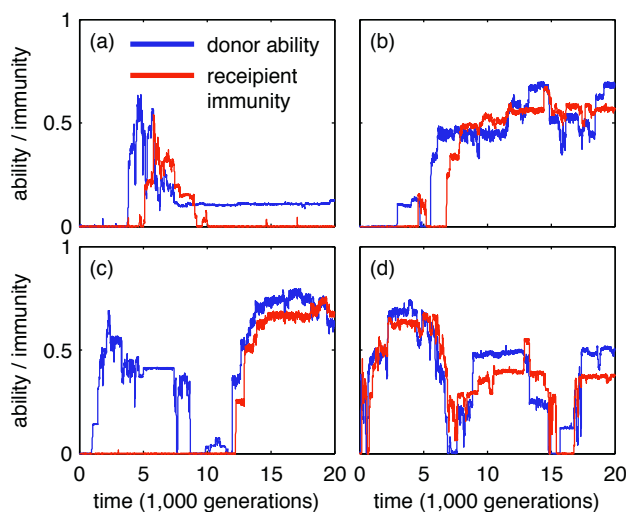
and potentially detrimental plasmids, it is the plasmid that is defending itself from being replaced. However, contrary to intuition of classical plasmid biology, our plasmids evolve to be larger than chromosome and also carry the majority of metabolic genes. Rather than being strictly selfish in their evolution and propagation, they also carry metabolic, directly beneficial genes that may transfer to future hosts. Still, the benefit is at best mutual, since by increasing the host organism's fitness the plasmid also effectively increases its vertical transmission rate and thus the probability of being transmitted into the next generation.



**Figure 4. Examples of the distribution of the fitness genes across genetic units.** We consider four replicate base experiments (no migration, no cost) over 20,000 generations of evolution (a-d). Different colors represent the average fitness measurements obtained when considering just the chromosome (green), just the plasmid (red), or the entire individual (blue).



However, while all experimental populations evolved to comparable fitness levels they have done so by following different trajectories in terms of the gene distribution between the chromosome and the plasmid. In Figure 4, we show some or the possible outcomes from 4 different populations, in which the metabolic genes are solely the plasmid (Figure 4a), on the chromosome (Figure 4b), are shared between the two during most of evolution (Figure 4c) or are constantly jumping between the genetic units (Figure 4d). In multiple panels of the Figure 4 it is obvious that the fitness measured for the plasmid and for the chromosome do not add up to the fitness of the individual. For example, between the generations 1,000 and 4,000 the average fitness of individuals in Figure 4b is much less than what would be expected based on the plasmid and chromosome fitness. Similar antagonistic or synergistic interactions are frequently observed and create an additional channel of interaction between the genetic units. The mechanistic explanation is a straightforward one, with obvious biological parallels. For example, both plasmid and the chromosome may express a gene that confers 0.4 level to a selected trait, raising their fitness when considered alone. However, if the optimal level of the trait is 0.6, the individual with both this chromosome and plasmid will overexpress this gene and may have lower fitness than expected from the contributions of its genetic units. Such interactions between the plasmid and chromosome are also a striking example of the benefits that come from models like *Aevol*, as they could



**Figure 5. Examples of the distribution of the transfer genes across genetic units.** We consider four replicate base experiments (no migration, no cost) over 20,000 generations of evolution (a-d). Different colors represent the average donor ability (blue) and recipient immunity (red) in the population. These examples are representative of the overall pattern of transfer rate evolution in the runs where both donor ability and recipient immunity evolved. Additionally, there were two experiments in which neither donor ability nor recipient immunity evolved (above an *ad hoc* threshold of 0.1 during at least 1,000 generations), as well as three in which donor ability evolved but recipient immunity never did.

not be observed in the classical, analytical models, and even here, they were not something we necessarily expected to see.

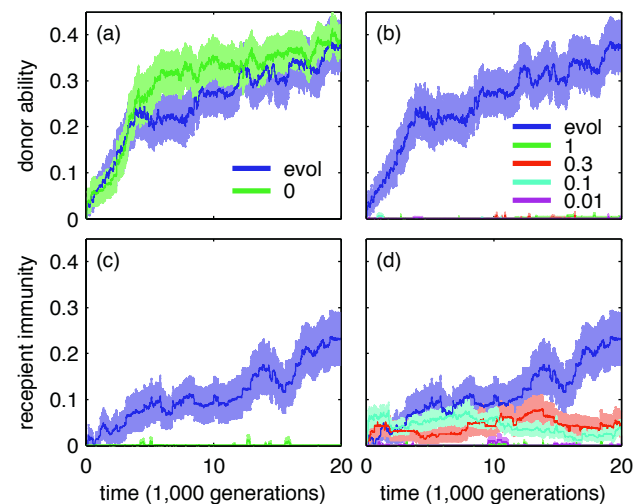
### Donor ability and recipient immunity coevolution

In order to examine the dynamics of plasmid transfer evolution, we examine the data from individual experiments, specifically the average plasmid donor ability and recipient immunity over the full course of the experiments. In Figure 5 we show four runs that are representative of the transfer rate evolution in our experiments and note two major trends:

(1) *Recipient immunity evolves only after donor ability.* This can be interpreted as an example of the apparent short-sidedness of evolution. Although immunity to invading plasmids is generally beneficial in the long run, without immediate benefit, any immunity genes are lost to drift.

(2) *A decrease or loss of donor ability is soon followed by the loss of recipient immunity.* As in the previous case, the recipient immunity without donor ability confers no benefit for the organism and is thus quickly lost by drift alone.

We suspect that in situations where recipient immunity is maintained at levels higher than donor ability, such as around generation 7000 in Figure 5a, the genetic architecture constraints relevant genes in a way that makes it difficult to decrease their expression levels without either a decrease in fitness or a complete loss of immunity.



**Figure 6. Evolution of transfer with fixed donor ability or recipient immunity.** (a, c) Comparison of the experiments with freely evolving recipient ability (dark blue line) and ones where the extrinsic recipient immunity was set to zero (green line). (b, d) Comparison of the experiments with freely evolving donor ability (dark blue line) and ones where the extrinsic donor ability was set to a fixed value of 1, 0.3, 0.1 or 0.01. Color legend in panel (a) is relevant for both (a) and (c), and the one in panel (b) for both (b) and (d). The lines are mean values for 20 replicate experiments within each treatment and the shaded area represents one standard error of the mean. The data for evolving donor ability and recipient immunity experiments (dark blue lines) are from the baseline runs, as in the previous figures.

To further assess the co-evolution of donor ability and recipient immunity we conducted two sets of experiments in which one of these traits was not allowed to evolve. When individuals had no possibility to modulate the recipient immunity (Figure 6a and 6c), the donor ability evolved to somewhat elevated, but not significantly higher levels. However, the average probability of transferring the plasmid (calculated just as the difference between the corresponding curves in Figure 6a and 6c) does differ between the two treatments by more than 60% and is significant (two-sided t-test,  $p < 0.001$ ). This indicated there is no optimal level of transfer ability that evolves, at least in the time frame of our study. Instead, the amount of plasmid transfer depends on the environment in which the plasmids are evolving in, and in this case, the ability of individuals to fight off unwanted plasmids.

Finally, we examined the evolution of recipient rate when the individuals cannot evolve their donor ability. In this case, setting the donor ability to zero would not provide any new or interesting outcomes, as all our previous results point to recipient immunity not increasing in the absence of donor ability. Instead, we set the external, unchanging donor ability and let the recipient ability evolve in response (Figure 6b and 6d). We found that recipient immunity does evolve at intermediate levels of base donor ability (0.1, 0.3), but not the extreme ones (1, 0.01). In neither case did the recipient immunity evolve to the levels as high as when donor ability also evolved freely (two-sample t-test,  $p < 0.01$ ). Although the range of constant, pre-set donor ability values was comparable to the ones that evolved freely, the recipient ability just could not compensate and “catch up”. We take this as another strong indication that the evolutionary fate of transfer rate is shaped by the interactions between donor ability and recipient immunity – only when immunity could co-evolve with donor ability, following it at a relatively short distance, it rose to higher values. A closer examination of the order in which mutations arose and spread, across a number of replicate populations, could provide the definite description of this coevolution, but at this time, remains extremely computationally demanding and outside of the scope of this study.

### Plasmid replacement and swapping

Bacterial conjugation is a form of sex, but compared to recombination, it is clearly one-sided and asymmetric: the flow of genetic information is unidirectional as there is a donor and a recipient of the plasmid. Motivated by plasmid retrotransfer, an exception to this rule, we modified the mechanism of conjugation in *Aevol* to swap two plasmids, rather than replace one with the copy of the other during every transfer event. We measured average donor ability (mean =  $1.9 \times 10^{-5}$ , standard error of the mean =  $7.0 \times 10^{-6}$ ) and recipient immunity (mean =  $4.1 \times 10^{-5}$ , standard error of the mean =  $1.7 \times 10^{-6}$ ) in the 20 replicate runs with swapping plasmid. Although both were significantly different than zero statistically ( $p = 0.015$  for donor ability,  $p = 0.027$  for recipient immunity, one-sample t-test) given the extremely low values, we do not consider them to be significant biologically but just a product of mutation-selection balance. We did hope to observe some digital sex, but given our previous results, these outcomes are not surprising. By swapping plasmids, we stopped them from being infectious,

as the frequency of a plasmid could not increase purely via horizontal transfer. Instead, the plasmid transfer-related processes were now closer to sex and recombination and thus subject to similar short-term costs and only long-term benefits. In absence of parasite-driven Red Queen dynamics (Lively et al. 1990), changing environment (Misevic et al. 2010), or one of the other scenarios beneficial for sexual reproduction (West et al. 1999) the organisms are likely to remain asexual, consistent with our results.

## Conclusions

Plasmids represent an important feature of microbial biology, have been extensively studied and used as an important tool in genetics, molecular and synthetic biology. However, many questions remain open, especially relating to plasmid evolution, interactions between the host and the plasmid, genetic architecture, and control over plasmid transfer. Here we presented an implementation of *in silico* transferable elements for the evolution platform *Aevol*. We demonstrate the strength of the approach by tackling classical research question of plasmid cost, but also investigate the evolutionary dynamics of metabolism and transfer genes in a way that would be extremely difficult to do in a natural system. We find signatures of coevolution between donor ability and recipient immunity, which evolve to be primarily, but not exclusively, encoded on the plasmid. Although plasmids seem to behave like selfish genetic elements, they at times also carry metabolic genes and may thus be directly beneficial to future hosts. Finally, during relatively long stretches of evolutionary time, the genes on the chromosome and the plasmid interacted epistatically, highlighting another way these genetic elements may affect each other's evolution. Throughout the study, the general trends were apparent, but were also accompanied by much stochastic variation and great diversity in the evolutionary trajectories. Future studies with *Aevol* will enable close analysis of individual-based effects as well as interactions between plasmid conjugation and other high-impact evolutionary events such as the evolution of cooperation or the evolution of multicellularity.

## References

- Alekshun MN and Levy SB. 2007. Molecular mechanisms of antibacterial multidrug resistance. *Cell* 128:1037-1050.
- Bennett PM. 2008. Plasmid encoded antibiotic resistance: acquisition and transfer of antibiotic resistance genes in bacteria. *Br J Pharmacol* 153:S347-S357.
- Bergstrom CT, Lipsitch M, and Levin BR. 2000. Natural selection, infectious transfer and the existence conditions for bacterial plasmids. *Genetics* 155:1505-1519.
- Beslon G, Parsons DP, Sanchez-Dehesa Y, Pena JN, and Knibbe C. 2010. Scaling laws in bacterial genomes: a side-effect of selection of mutational robustness. *Biosystems* 102:32-40.
- Cohen SN, Chang AC, Boyer HW, and Helling RB. 1973. Construction of biologically functional bacterial plasmids in vitro. *Proc Natl Acad Sci USA* 70:3240-3244.

- Connelly BD, Zaman L, McKinley PK, and Ofria C. 2011. Modeling the evolutionary dynamics of plasmids in spatial populations. Pp. 227-233 in N Krasnogor, and PL Lanzi, eds. Proceedings of Genetic and Evolutionary Computation Conference (GECCO), Dublin, Ireland.
- Constante M, Gruenberg R, and Isalan M. 2011. A biobrick library for cloning custom eukaryotic plasmids. PLoS ONE 6.
- Davison J. 1999. Genetic exchange between bacteria in the environment. Plasmid 42:73-91.
- Doolittle WF. 1999. Phylogenetic classification and the universal tree. Science 284:2124-2128.
- Goldsby HJ, Dornhaus A, Kerr B, and Ofria C. 2012. Task-switching costs promote the evolution of division of labor and shifts in individuality. Proc Natl Acad Sci USA 109:13686-13691.
- Knibbe C, Coulon A, Mazet O, Fayard J-M, and Beslon G. 2007. A long-term evolutionary pressure on the amount of non-coding DNA. Mol Biol Evol 24:2344-2353.
- Knibbe C, Fayard J-M, and Beslon G. 2008. The topology of the protein network influences the dynamics of gene order: from systems biology to a systemic understanding of evolution. Artificial Life 14:149-156.
- Knibbe C, Mazet O, Chaudier F, Fayard J-M, and Beslon G. 2006. Evolutionary coupling between the deleteriousness of gene mutations and the amount of non-coding sequences. J Theor Biol 244:621-630.
- Koonin EV, Makarova KS, and Aravind L. 2001. Horizontal gene transfer in prokaryotes: Quantification and classification. Annu Rev Microbiol 55:709-742.
- Koonin EV and Wolf YI. 2012. Evolution of microbes and viruses: a paradigm shift in evolutionary biology? Front Cell Infect Microbiol 2:119-119.
- Krone SM, Lu R, Fox R, Suzuki H, and Top EM. 2007. Modeling the spatial dynamics of plasmid transfer and persistence. Microbiology 153:2803-2816.
- Lenski RE, Ofria C, Collier TC, and Adami C. 1999. Genome complexity, robustness and genetic interactions in digital organisms. Nature 400:661-664.
- Levin BR and Stewart FM. 1980. The population biology of bacterial plasmids: *a priori* conditions for the existence of mobilizable nonconjugative factors. Genetics 94:425-443.
- Lipps G, ed. 2009. Plasmids: Current Research and Future Trends. Caister Academic Press, Norfolk, UK.
- Lively CM, Craddock C, and Vrijenhoek RC. 1990. Red Queen Hypothesis supported by parasitism in sexual and clonal fish. Nature 344:864-866.
- Misevic D, Frénoy A, Parsons DP, and Taddei F. 2012. Effects of public good properties on the evolution of cooperation. Pp. 218-225 in C Adami, DM Bryson, C Ofria, and RT Pennock, eds. Proceedings of Artificial Life 13, East Lansing, MI.
- Misevic D, Ofria C, and Lenski RE. 2004. Sexual reproduction and Muller's ratchet in digital organisms. Pp. 340-345 in JB Pollack, M Bedau, P Husbands, T Ikegami, and RA Watson, eds. Proceedings of Artificial Life IX. MIT Press, Cambridge, Massachusetts.
- Misevic D, Ofria C, and Lenski RE. 2006. Sexual reproduction reshapes the genetic architecture of digital organisms. Proc R Soc Lond B 273:457-464.
- Misevic D, Ofria C, and Lenski RE. 2010. Experimental evidence for evolution of sex in changing environments J Hered 101:S46-S54.
- Ochman H, Lawrence JG, and Groisman EA. 2000. Lateral gene transfer and the nature of bacterial innovation. Nature 405:299-304.
- Parsons DP, Knibbe C, and Beslon GTC. 2010. Importance of the rearrangement rates on the organization of transcription. Pp. 479-486 in H Fellermann, M Dörr, MM Hanczyc, L Ladegaard Laursen, S Maurer, D Merkle, P-A Monnard, K Stoy, and S Rasmussen, eds. Proceedings of ALife XII. MIT Press, Cambridge, MA.
- Pray L. 2008. Recombinant DNA technology and transgenic animals. Nature Education 1.
- Ragan MA, McInerney JO, and Lake JA. 2009. The network of life: genome beginnings and evolution. Philos Trans R Soc B Biol Sci 364:2169-2175.
- Sambrook J, Fritsch EF, and Maniatis T. 1989. Molecular Cloning: A Laboratory Manual. Cold Spring Harbor Laboratory Press, New York.
- Sia EA, Kuehner DM, and Figurski DH. 1996. Mechanism of retrotransfer in conjugation: Prior transfer of the conjugative plasmid is required. J Bacteriol 178:1457-1464.
- Smillie C, Pilar Garcillan-Barcia M, Victoria Francia M, Rocha EPC, and de la Cruz F. 2010. Mobility of Plasmids. Microbiol. Mol. Biol. Rev. 74:434-452.
- Stewart FM and Levin BR. 1977. The population biology of bacterial plasmids : *a priori* conditions for the existence of conjugationally transmitted factors. Genetics 87:209-228.
- Svara F and Rankin DJ. 2011. The evolution of plasmid-carried antibiotic resistance. BMC Evol Biol 11.
- Szpirer C, Top E, Couturier M, and Mergeay M. 1999. Retrotransfer or gene capture: a feature of conjugative plasmids, with ecological and evolutionary significance. Microbiology-UK 145:3321-3329.
- West SA, Lively CM, and Read AF. 1999. A pluralist approach to sex and recombination. J Evol Biol 12:1003-1012.
- Zatyka M and Thomas CM. 1998. Control of genes for conjugative transfer of plasmids and other mobile elements. FEMS Microbiol Rev 21:291-319.

# Construction of a remote-controlled supramolecular micro-crawler

Daiki Komatsu<sup>1</sup>, Kei Fujiwara<sup>1</sup>, and Shin-ichiro M. Nomura<sup>1</sup>

<sup>1</sup>Tohoku University, 6-6-01 Aramaki Aza-Aoba, Aobaku, Sendai 980-8579, Japan  
komatsu@molbot.mech.tohoku.ac.jp, nomura@molbot.mech.tohoku.ac.jp

## Abstract

We aimed to develop a micro-robot that can crawl on contact surfaces in biological environments. The prototype chassis of this micro-bot consists of a lipid membrane that encapsulates and bonds micro-sized magnetic particles. By applying a rotating magnetic field, we hope to obtain a micro-crawler robot. In this report, we describe our observations of the rotational movement of liposomes (10–60  $\mu\text{m}$  in diameter) encapsulating magnetic particles following manipulation of an external magnetic field using a neodymium magnet. Since this robot actively makes contact with the external environment, it will be possible to salvage some important molecule from the contact surface. It is expected that development of this system will lead to the development of new diagnostic and treatment systems.

## Introduction

Since the dawn of history, numerous functional molecules have been discovered and synthesized by scientists. System integration of such molecular devices can facilitate the construction of human-controllable molecular machines. Designing and controlling nano- and micro-meter-sized chemical systems is considered one of the most effective means of examining the invisible small world. Recently, molecular self-organization has been gaining increasing interest with a view to creating higher-order chemical systems at the single-molecule level. This represents a crucial step in the development of the new research field of “Molecular Robotics” [1]. For prototyping a molecular robot, compartmentalization in a homogeneous aqueous solution is an essential requirement. One of the key materials in this regard is a supramolecular structure called the “liposome.” Liposomes consist of a closed phospholipid bilayer membrane and behave as hydrophilic capsules. These lipid vesicles have the property of excellent biocompatibility, are capable of holding various solutions, have readily modified surfaces, and can potentially be prepared in large amounts. Since their discovery, many applications, including carriers of drugs, have been studied.

The liposome “capsule” is essentially floating in a solution. By encapsulating magnetic particles within liposomes, active drug delivery to target tissues could be realized by controlling the external magnetic field. In this regard, studies of the positional control of liposomes in blood vessels using MRI apparatus have been reported [2]. Local accumulation of liposomes in the fluid environment and the control of drug release by applying external stimuli using a high-temperature superconducting bulk magnet have also been performed [3]. However, such floating particles are unable to detect surface

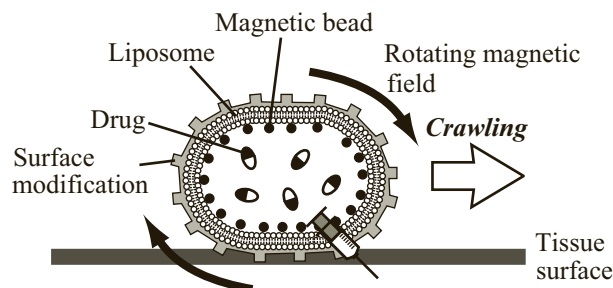


Figure 1: Schematic image of supramolecular micro-crawler.

molecular information. Nevertheless, we have noted that investigations of the interactions between solutions and living surfaces is important for an understand of living systems.

To this end, we have attempted to construct a crawler-type micro-molecular robot (supramolecular micro-crawler; Figure 1). The liposomes of this crawler consist of lipid and adhered-encapsulated magnetic micro-particles. This multicomponent structure is expected to function on tissue in the body environment, for example, via vascular flow, by crawling induced by external rotating magnetic field.

We believe that these types of robots will not only have the ability of remotely controllable drug delivery in three dimensional motion, but will also be able to determine normal or abnormal areas of tissue by sensing the surface molecules through interaction with the cellular-contact surface. We also anticipate the construction of new diagnostic and therapeutic systems, such as the robot described here, and that these will be applied to the treatment of malignant tissue through making a diagnosis of living tissue as a result of rolling on the tissue surface.

In this paper, we present the results of the rotational motion of liposome-encapsulated magnetic beads generated by applying an external magnetic field to construct the micro-robot described above.

## Experiments

We adopted the water-in-oil (W/O) emulsion method to prepare the liposomes (Figure 2) [4]. The composition of the buffer solution used was as follows: 10 mM HEPES-KOH, 2.6 mM  $\text{Mg}(\text{OAc})_2$ , with 20 mM potassium glutamate (pH 7.6). A lipid mixture of 1,2-dioleoyl-sn-glycero-3-phosphoethanolamine-n-(biotinyl) (sodium salt) (biotinyl DOPE), 1,2-dipalmitoyl-sn-glycero-3-phosphocholine



(DPPC), and cholesterol (at molar ratio of 1:1:1) was dissolved in liquid paraffin.

The lipid solution was mixed with the buffer containing 150 mM sucrose, 350 mM glucose and streptavidin magnetic particles (inner solution). This mixture was vortexed for 60 s to form a W/O emulsion. The emulsion was then gently placed on top of the buffer containing 500 mM glucose (outer solution) in a tube. The sample tube was centrifuged and the emulsion was then passed through an oil/water interface

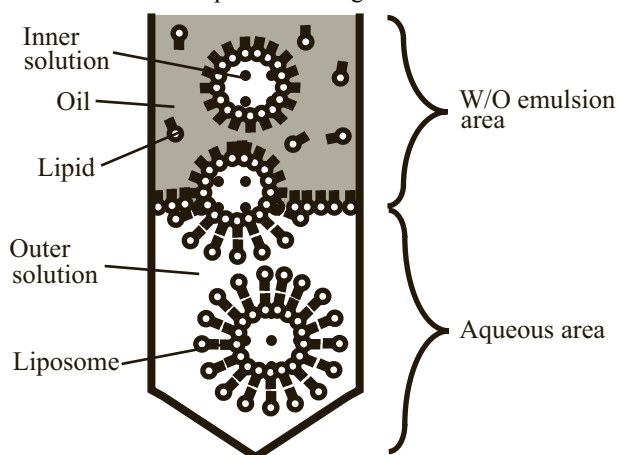


Figure 2: The W/O emulsion method for liposome formation.

saturated with lipids to form a bilayer structure. The procedure is a slightly modified version of that previously described by Nishijima et al. [5].

The top-most W/O emulsion was removed and the liposomes (maximum diameter of approximately 60  $\mu\text{m}$ ) encapsulating magnetic beads were collected by micropipette. The liposomes thus obtained were clearly observable under a phase-contrast microscope (IX-71; Olympus). On the microscope stage, we attempted to observe the rotational movement of liposomes containing the magnetic particles by applying an external magnetic field. The rotational magnetic field was generated using a round-type neodymium magnet ( $\phi$  11 mm  $\times$  3 mm). Sample solution containing liposomes was placed in a hole in a silicone sheet on a slide glass, and covered with a cover glass. The magnet was placed next to the prepared slide on the microscope stage.

## Results and Discussion

The rotating magnetic field was generated in two different ways: (A) by moving the neodymium magnet around the samples, and (B) by rotating the magnet near the samples. The results for each method are shown in Figure 3A and B, respectively. It was possible to perform rotational movement using both methods. The rotational cycle was not fast and limited to  $\sim 0.3$  Hz. Surface resistance between the double-layer of the liposome membrane, and viscosity of the buffer solution should be affected. We then attempted to place the suitable chemical structure onto the liposome surface. The results of this investigation will be discussed in the conference.

In this study, we controlled liposome movement by applying a magnetic field to liposomes encapsulating magnetic particles for the construction of a supramolecular micro-crawler, which will be used to develop new diagnostic and treatment systems, and we succeeded in obtaining rotational motion of the our designed liposomes using two different rotational magnetic fields. We are currently examining the crawling motion of liposomes on surfaces. For this purpose, we will modify the surface properties of the

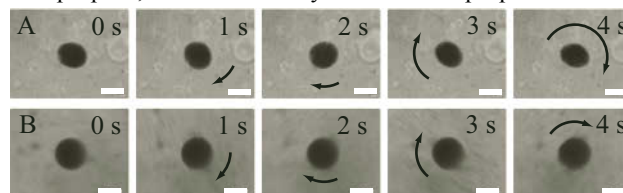


Figure 3: Rotational behaviors of a liposome using two different methods. Scale bars are 20  $\mu\text{m}$ .

liposomes and the contact surface (e.g., molecular modification and charge), and conduct experiments in static and fluid environments. In addition, we are constructing a device that will be used to rotate the liposomes. We also aim to collect target molecules consisting of fluorescent molecules using a micro-robot prepared from liposomes.

## Acknowledgements

We would like to thank Prof. S. Murata and S. Hamada for their insightful comments. This work was supported by JSPS KAKENHI (grant numbers 24104004, 22220001 and 23.3718).

## References

- [1] Murata, S., Konagaya, A., Kobayashi, S., Saito, H. and Hagiya, M. (2013). Molecular Robotics: A New Paradigm for Artifacts. *New Generation Computing*, 31:27-45.
- [2] Martel, S. (2012). Magnets steer medical microbots through blood vessels. *IEEE Spectrum*, 49(10):48-53.
- [3] Nakagawa, K., Mishima, F., Akiyama, Y. and Nishijima, S. (2012). Study on Magnetic Drug Delivery System Using HTS Bulk Magnet. *IEEE Transactions on Applied Superconductivity*, 23(3): 4903804.
- [4] Pautot, S., Frisken, J. B. and Weitz, D. A. (2003). Engineering asymmetric vesicles. *Proceedings of the National Academy of Sciences of the United States of America*, 100(19): 10718-10721.
- [5] Nishijima, K., Hosoi, T., Sunami, T., Toyota, T., Fujinami, M., Oguma, K., Matsuura, T., Suzuki, H. and Yomo, T. (2009). Population Analysis of Structural Properties of Giant Liposomes by Flow Cytometry. *Langmuir*, 25(18): 10439-10443.



# Evolution of G-P mapping in a von Neumann Self-reproducer within *Tierra*

Declan Baugh, Barry McMullin

The Rince Institute, Dublin City University, Ireland

[declan.baugh2@mail.dcu.ie](mailto:declan.baugh2@mail.dcu.ie), [barry.mcmullin@dcu.ie](mailto:barry.mcmullin@dcu.ie)

## Abstract

John von Neumann first presented his theory of machine self-reproduction in the late 40's (von Neumann, 1948), in which he described a machine capable of performing the logical steps necessary to accommodate self-reproduction. The proposed architecture was comprised of two distinct components, a passive genotype, which acts exclusively as an information storage of a machine description, and an active phenotype which is responsible for all mechanical functionality of the machine including the ability to decode the genotype and construct the described machine to facilitate self-reproduction. This paper presents an exploratory model which implements the von Neumann architecture for self-reproduction within the pre-existing evolutionary platform of *Tierra*. Initially, the memory image of the automaton's genotype and phenotype are physically identical, and each symbol in memory may be interpreted as either passive numerical data (g-symbol), or a functional instruction (p-symbol), depending on how the symbol is interpreted. If redundancy is introduced to a mutable genotype-phenotype mapping, the mapping system becomes non-invertible, rendering it impossible to compute an automaton's exact genotypic memory image by analysis of the phenotype alone. However, this non-invertible mapping may allow for a more robust genotype, increasing its robustness to fatal mutations and therefore increasing its ability to preserve its phenotypic form under perturbations.

## The von Neumann Architecture for Machine Self-reproduction

Von Neumann's architecture for machine self-reproduction, presented in his theory of self-reproducing automata (von Neumann, 1966; Baugh and McMullin, 2012), describes a machine,  $M$ , which is decomposed into two primary components, a functional component  $P$ , and an passive component  $G$ , such that  $M = (P + G)$  (McMullin, 2012).  $G$  represents a one-dimensional string of symbols which has no active/functional capability, but can be interpreted as information, similar to the *tape* of a Turing machine. The information within  $G$  is used to describe an arbitrary machine  $X$  under some function,  $\phi()$ , such that  $G = \phi(X)$ .

$P$  is further divided into four fundamental subcomponents, a *general constructive automaton*  $A$ , a *general copy-*

*ing automaton*  $B$ , a *control unit*  $C$ , and the *ancillary machinery*,  $D$ .  $G$  will be referred to as the *genotype* while  $P$  will be referred to as the *phenotype*<sup>1</sup>.

The general constructive automaton  $A$  can read the symbols within  $G$ , and interpret them as an encoded description of an arbitrary machine  $X$ .  $A$  has the capability to apply an inverse function,  $\phi^{-1}()$ , or  $\psi()$ , to  $G$ , and construct the described machine  $X$ . We denote this by saying  $\psi(G) = \psi(\phi(X)) = \phi^{-1}(\phi(X)) = X$ . In other words, when supplied with a genotype, the general constructive automaton applies the decoding function  $\psi()$ , to  $G$ , in order to construct the arbitrary machine  $X$ .

The general copying automaton  $B$ , reads and duplicates the machine description  $\phi(X)$ . A control unit  $C$  is required to govern the automaton ( $A + B$ ), directing its operation, activating  $A$  and  $B$  in the correct order, and insuring that the offspring creature is "activated" once its construction is complete.

The forth component, the ancillary machinery  $D$ , refers to all conceivable functionality that the machine may possess which does not interfere or hinder the reproductive operation of ( $A + B + C$ ).

When a machine ( $A + B + C + D$ ) is supplied with a description  $\phi(X)$ , the control unit  $C$  first commands  $B$  to duplicate  $\phi(X)$ . Upon duplication,  $C$  instructs  $A$  to decode  $\phi(X)$  under some inbuilt genotype-phenotype mapping function  $\psi()$ , and construct the described machine  $X$ . Finally  $C$  will attach the new instances of  $\phi(X)$  and  $X$ , and sever them from the parent automaton ( $A + B + C + D$ ), after which there exists the new entity,  $X + \phi(X)$ . Now consider the case where  $X = (A + B + C + D)$ . This system,  $(A + B + C + D) + \phi(A + B + C + D)$  will proceed to construct an offspring automaton and attach it to the description of itself,  $(A + B + C + D) + \phi(A + B + C + D)$ . The parent and offspring are identical, therefore achieving self-reproduction. This machine architecture is demonstrated in Figure 1.

<sup>1</sup>Although von Neumann never used these terms, we now associate the components in question with the genotype and phenotype in organic biology.

Next we consider the case where a random phenotypic perturbation occurs during the construction of  $P$ , affecting  $D$ , so that a machine  $M = (P + G)$  produces  $M' = (P' + G)$ , where  $P' = (A + B + C + D')$ . This machine  $(A + B + C + D') + \phi(A + B + C + D)$  will proceed to decode and copy the unaltered  $G$  under a decoding function  $\psi()$ , to recreate the original machine  $M = (A + B + C + D) + \phi(A + B + C + D)$ , and the phenotypic perturbation is not inherited. For the case where the perturbation affects the description of  $D$  in  $G$ , creating a machine  $M' = (A + B + C + D) + \phi(A + B + C + D')$ ,  $M'$  will proceed to decode and copy  $G'$  under a decoding function  $\psi()$  to create a new machine,  $M'' = (A + B + C + D') + \phi(A + B + C + D')$ , where  $M'' = (P' + G')^2$ .

Should a random perturbation occur when copying the description of  $A$  to an offspring, which results in  $\phi(A' + B + C + D)$ , then the machine  $(A + B + C + D) + \phi(A' + B + C + D)$  will produce  $(A' + B + C + D) + \phi(A' + B + C + D)$ . It is possible that this machine will now have an altered general constructive automaton. Von Neumann/Burks stated “If the change is in  $A$ ,  $B$  or  $C$ , the next generation will be sterile.” (von Neumann, 1966, p. 86), however it is conceivable that a perturbation within the description of  $A$  may only affect the decoding function  $\psi()$  without completely breaking its reproductive functionality.

This machine  $(P' + G')$  may not be sterile, but conduct the altered decoding function  $\psi^*()$  where  $\psi^*(G') = \psi^*(\phi(P')) = (P')^*$ , to construct the machine  $((P')^* + G')$ . If there are changes in the mapping which allow  $(P')^* = P'$ , then the machine  $(P' + G')$  will self-reproduce successfully while conducting a different genotype-phenotype mapping (McMullin, 2000). We aim to scrutinise this intricate detail, and investigate if any additional mutational pathways may emerge as a result of a mutable genotype-phenotype mapping.

## Implementation within the Tierra Platform

Tierra is an artificial life platform where populations of assembler language self-reproducing automatons (creatures) compete with one another within a one-dimensional circular core memory for both CPU time and memory space (Ray, 1991). Each Tierran automaton consists of a CPU with up to six registers, stack memory and an instruction pointer.

Typically, self-reproduction within Tierra is accomplished via *self-copying*, where a creature must inspect its entire memory image in order to construct an identical offspring. This mechanism is loosely analogous to the reproduction process which occurs in the RNA world hypothesis which posits that at the earlier stages of evolution, RNA acted as

<sup>2</sup>It is worth noting that when a perturbation occurs within  $P$ , the perturbation is not inherited in further generations, however when the perturbation occurs within  $G$ , there is a generation delay between when the perturbation occurs in the genotype and when it is expressed in the phenotype.

both template and template-directed polymerase, and there existed no distinction between genotype and phenotype.

In order to implement the von Neumann architecture within the platform of Tierra, the seed automaton must enforce a division of labour between the storage of genetic information and the catalytic functionality, hence recognising the roles of genotype and phenotype.

The phenotype will naturally consist of the three sub-components, a general constructive automaton  $A$ , a general copying automaton  $B$ , and a control unit  $C$ . The control unit segment of the automaton will calculate the offspring size and allocate memory space to construct the offspring. The general constructor segment will incorporate a mutable genotype-phenotype mapping to allow for inheritable variation which may result in new evolutionary trajectories with creatures conducting an altered genotype-phenotype mapping. The general constructor will incrementally read the symbols within the genotype and under some genotype-phenotype mapping, will determine which p-symbols are to be written to the offspring phenotype. Upon construction of the offspring phenotype the copier is activated and the g-symbols within the parent's genotype are incrementally read and written to the offspring genotype. The control unit then activates the offspring automaton, and the reproductive cycle repeats.

## mRNA-Amino Acid Inspired Genotype-phenotype Mapping

In order to encode the phenotype of an automaton, an arbitrary genotype-phenotype mapping must be implemented. The evolutionary trajectory of such an automaton will be in part, determined by the nature of this arbitrarily elected mapping. We can only claim that any phenomenon observed will be specific and characteristic to the specific mapping system which is implemented. For the purpose of this project, a bijective, mono-alphabetic substitution cipher was chosen. This method was loosely based on the genetic code, in which an mRNA, consisting of a one-dimensional string of symbols (nucleotides), is transcribed into a specific string of symbols (amino acids). If a single letter in an mRNA codon gets perturbed, then the affected codon may result in the construction of a different amino acid.

If we implement a similar mapping system which allows perturbations to the genotype which may alter the description of the general constructor, specifically, altering the genotype-phenotype mapping function  $\psi()$ , then we would effectively be implementing a von Neumann reproducer which may give rise to new evolutionary trajectories operating an altered genotype-phenotype mapping.

This type of a mutable genotype-phenotype mapping was facilitated via the inclusion of a lookup table within the general constructor. The lookup table consists of a one-dimensional string representing the full list of p-symbols available within the phenotype space.

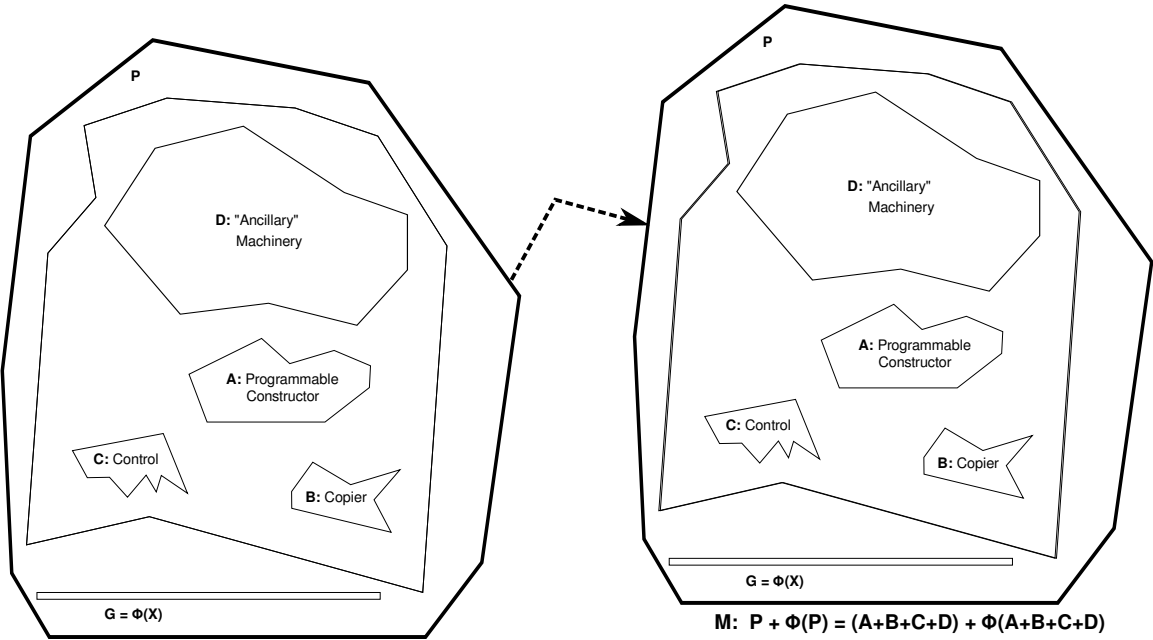


Figure 1: A schematic of the von Neumann style architecture of machine self-reproduction. Excerpted from McMullin (2012).

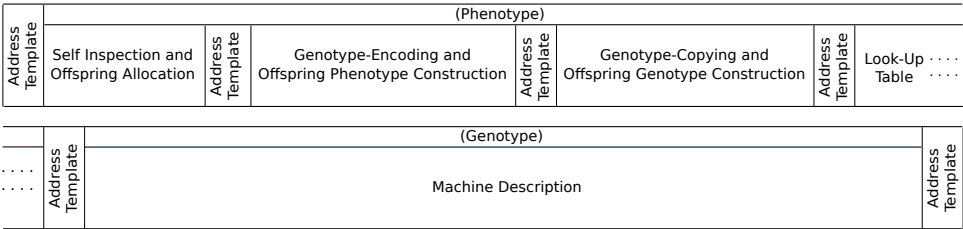


Figure 2: A schematic of a von Neumann style ancestor in Tierra. The genotype-encoding and offspring phenotype construction, and the lookup table comprises the general constructive automaton. The genotype-copying and offspring genotype construction segment comprises the general copying automaton, and the self-inspection and offspring allocation segment comprises the control unit.

During construction of an offspring phenotype, the parent's genotype is incrementally examined by the general constructor. The g-symbol at a memory location is read and interpreted as its underlying binary numerical value. The p-symbol situated at the corresponding relative address in the lookup table is read, and is written to the offspring's phenotype where it will function as an active instruction. The general constructor sequentially executes this process for every location in the genotype in order to decode each g-symbol and construct the offspring phenotype.

In an attempt to replicate conditions early in the phase change from simple RNA replicators to a system of mRNA and amino acids, we implemented an identity mapping from genotype to phenotype, where the content of the one-dimensional memory image of the genotype is physically identical to that of the phenotype. This situation is analogous to that of the RNA world hypothesis, where in order to replicate, an RNA molecule can act as both a functional catalyst, or a string of symbols to be interpreted, depending on whether it is acting as the catalyst or template.

In cryptography we would refer to this identity mapping as being an encryption in which the plaintext is identical to the ciphertext. Plaintext is information a sender wishes to transmit to a receiver and ciphertext is the result of encryption performed on the plaintext using an algorithm, called a cipher. In this case, the phenotype can be thought of as an un-encrypted message which is to be transmitted to the offspring's memory image and the genotype is the ciphertext which was a result of encrypting the phenotype, according to the encoding function  $\phi()$ . The initial decoding function  $\psi()$  is determined by the permutation of symbols within the lookup table. It is worth noting however, that regardless of the initial permutation chosen for the lookup table, the initial description of the lookup table will always remain the same. If we let  $S$  represent the non-permuted list of symbols which exist in the Tierra universe<sup>3</sup>, then the permutation of the lookup table depicts how each individual element within this non-permuted list of symbols is decoded under  $\psi()$ <sup>4</sup>. The lookup table can now be described as  $\psi(S)$ . When we encode a phenotype to acquire the genotype, the encoded lookup table will now be represented by  $\phi(\psi(S)) = \psi^{-1}(\psi(S)) = S$ , therefore, regardless of the initial permutation of the lookup table, the initial description of the lookup table will always take the form of  $S$ .

Our ancestor requires a minimum of 28 phenotypic instructions in order to self-reproduce. The mRNA-amino acid transcription table consists of a genotype space of 64 different codons, but a significantly smaller phenotype space of

<sup>3</sup>In this case, the non-permuted list of symbols is represented by a list of consecutive binary numbers from 000000 to 111111.

<sup>4</sup>The first location in the lookup table represents which p-symbol is mapped onto by 000000. The second position in the lookup table represents which p-symbol is mapped onto by 000001 etc.

UUU	(Phe/F)	UCU	(Ser/S)	UAU	(Tyr/Y)	UGU	(Cys/C)
UUC		UCC		UAC		UGC	
UUA		UCA		UAA	Stop(Ochre)	UGA	Stop(Opal)
UUG		UCG		UAG	Stop(Amber)	UGG	(Trp/W)
CUU	(Leu/L)	CCU		CAU	(His/H)	CGU	
CUC		CCC	(Pro/P)	CAG		CGC	(Arg/R)
CUA		CCA		CAA	(Gln/Q)	CGA	
CUG		CCG		CAG		CGG	
AUU		ACU		AAU	(Asn/N)	AGU	(Ser/S)
AUC	(Ile/I)	ACC	(Thr/T)	AAC		AGC	
AUA		ACA		AAA	(Lys/K)	AGA	(Arg/R)
AUG	(Met/M)	ACG		AAG		AGG	
GUU		GCU		GAU	(Asp/D)	GGU	
GUC	(Val/V)	GCC	(Ala/A)	GAC		GGC	(Gly/G)
GUA		GCA		GAA	(Glu/E)	GGA	
GUG		GCG		GAG		GGG	
00000	(nop0)	10000	(decC)	100000	(popC)	110000	(movAb)
00001	(nop1)	10001	(nop10)	100001	(nop19)	110001	(nop29)
00010	(nop2)	10010	(incD)	100010	(popD)	110010	(movda)
00011	(nop3)	10011	(nop11)	100011	(nop20)	110011	(nop30)
00100	(ifnz)	10100	(pushA)	100100	(popE)	110100	(movAb)
00101	(nop4)	10101	(nop12)	100101	(nop21)	110101	(nop31)
00110	(addAAE)	10110	(nop13)	100110	(nop22)	110110	(ret)
00111	(nop5)	10111	(nop14)	100111	(nop23)	110111	(nop32)
01000	(subCAB)	11000	(pushC)	101000	(jmpb)	111000	(nop33)
01001	(nop6)	11001	(nop15)	101001	(nop24)	111001	(shlA)
01010	(subAAC)	11010	(pushD)	101010	(adrf)	111010	(nop34)
01011	(nop7)	11011	(nop16)	101011	(nop25)	111011	(nop35)
01100	(incA)	11100	(popA)	101100	(nop26)	111100	(mal)
01101	(nop8)	11101	(nop17)	101101	(nop27)	111101	(nop36)
01110	(incB)	11110	(popB)	101110	(call)	111110	(nop37)
01111	(nop9)	11111	(nop18)	101111	(nop28)	111111	(divide)

Figure 3: The upper figure presents the mapping from mRNA to amino acid. The lower figure presents the initial mapping from g-symbols to p-symbols, implemented with our von Neumann style seed automaton. Red symbols highlight those which are non-employed and grey highlights symbols which initially map onto employed p-symbols.

22 amino acids, plus the start and stop codons. In an attempt to mirror the redundancy of the genetic code, a genotype and phenotype space of 64 was implemented. This corresponds to 64 separate 6-bit binary digits in the genotype space, and 64 phenotypic instructions, 28 of which have an active function and are used within the phenotype and contribute towards self-reproduction, and 36 of which have no active function at all. (See Figure 3.)

## Experimental Procedure

The Tierra soup was inoculated with the described von Neumann style ancestor, (Figure 2).

Point perturbations which affect random memory locations throughout the soup (cosmic rays) and perturbations which occur exclusively to symbols that are being written to memory locations in the soup (copy perturbations) were enabled and the system was run for 100 billion CPU cycles, which is approximately 250 thousand generations<sup>5</sup>. Every strain of creature that emerged throughout the run was captured and the number of employed and non-employed p-symbols within the lookup table for each creature was

<sup>5</sup>A generation in Tierra is a calculated time interval, which is determined by the estimated average amount of CPU cycles required for each creature present in the soup to reproduce once and die.



counted. Employed p-symbols refer to those which have a functional role in the process of reproduction, while non-employed p-symbols are included to introduce redundancy and do not actively contribute towards the reproduction process. If a specific p-symbol exists in the lookup table, then there must exist a specific g-symbol which maps onto it. If a p-symbol is absent from the lookup table then it is lost from the genotype-phenotype mapping. With our current mapping system<sup>6</sup>, it is impossible for a p-symbol which is absent in a parents lookup table to be included in its offspring's phenotype (with the exception of random phenotypic perturbations introducing random p-symbols to an offspring).

The population of employed vs. non-employed p-symbols in the lookup table of each creature was then plotted against the time of emergence of that individual, and this process was repeated 4 times.

## Results

### Standard evolutionary Behaviour

Initial simulations showed evolutionary behaviour similar to that documented in Ray's initial experiments (Ray, 1991). Informational parasitism<sup>7</sup> quickly emerged due to the description of the lookup table being omitted from the genotype. The resulting creature will redirect its CPU to a neighbouring host to facilitate the construction of its phenotype and therefore expend less CPU time per reproduction cycle due to its reduced length. Another evolutionary phenomena typical of Rays experiments is the reduction of creature size by reducing template addresses where possible. Address locations in Tierra are not facilitated via a global address location, but by matching complementary patterns of `nop1`'s and `nop0`'s. While the programmer creating the creature may use an initial template size of four `nop` instructions, evolution will typically reduce the template size where ever possible, creating shorter and more efficient offspring. Other evolutionary behaviours observed when implementing von Neumann style reproduction are the emergence of pathological constructors (Baugh and McMullin, 2012) and the degeneration to self-copiers within in the platform of Avida (Hasegawa and McMullin, 2012).

### Evolution of the Genotype-phenotype mapping

The aforementioned evolutionary behaviours have already been studied and documented, and therefore is not of primary concern, so for the remaining experiments the input parameters were edited so that only offspring of the same length as the initial ancestor are allowed to propagate throughout the memory. This will prevent the distraction of known phenomenon occurring and allow us to fo-

cus on the specific evolutionary lineages which arise as a direct result of a change in the genotype-phenotype mapping. A change in the genotype-phenotype mapping will be most easily recognised by a change in the lookup table.

Initially, non-fatal inheritable silent perturbations<sup>8</sup> of the genotype will occur in the description the lookup table. This will alter the genotype-phenotype mapping and allow previously silent g-symbols<sup>9</sup> to be mapped onto employed p-symbols. This allows single employed p-symbols to be mapped onto by multiple g-symbols.

The initial ancestor has 36 silent g-symbols, which are mapped onto 36 different non-employed p-symbols. As neither the silent g-symbols nor the non-employed p-symbols functionally contribute to the reproduction of offspring, the silent mutations that affect which p-symbol the silent g-symbols are mapped onto are random and arbitrary. However it was found that there was a strong bias towards the mapping of silent g-symbols onto employed p-symbols. During an evolutionary run, we see a sharp decrease in the number of non-employed p-symbols within the lookup tables of newly emerging creatures. Eventually, all 36 non-employed p-symbols are eliminated from the descendants of the initial ancestor, and the 64 positions in their lookup tables will consist almost entirely of employed p-symbols. This result can be seen in Figure 4.

## Discussions

The evolution of the genotype-phenotype mapping will initially be driven predominantly by the underlying physical dynamics of the coding system. The nature of the substitution cypher mapping mechanism employed means that certain perturbations of the lookup table are not directly reversible. This results in a biased drift in the genotype-phenotype mapping, eventually eliminating all non-employed p-symbols from the phenotype by ensuring that they are not mapped onto by any elements of the genotype space.

Figure 5 demonstrates a small section of the lookup table and its description. By studying a creature's lookup table and the lookup table's description, one can deduce the genotype-phenotype mapping that is implemented by that creature. For this small section of the mapping between the g-symbols and p-symbols, we see a set of four g-symbols which are interpreted as 0, 1, 2 and 3, which are mapped onto four p-symbols which are interpreted as `nop0`, `nop1`, `nop2` and `nop3` respectively. The red symbols within the lookup table represent non-employed p-symbols, while the grey symbols within the lookup table's description represent the silent g-symbols which initially map onto a non-employed p-symbol. Figure 5(a) demonstrates the initial 1st

<sup>6</sup>A mono-alphabetic substitution cipher.

<sup>7</sup>Informational parasitism refers to a form of parasitism which accesses and reads a host's memory contents, but does not directly interfere with its functionality.

<sup>8</sup>A silent perturbation is one which alters the genotypic sequence, but does not affect the structure of the phenotype.

<sup>9</sup>By silent g-symbols, we refer to g-symbols which were initially mapped onto non-employed p-symbols.



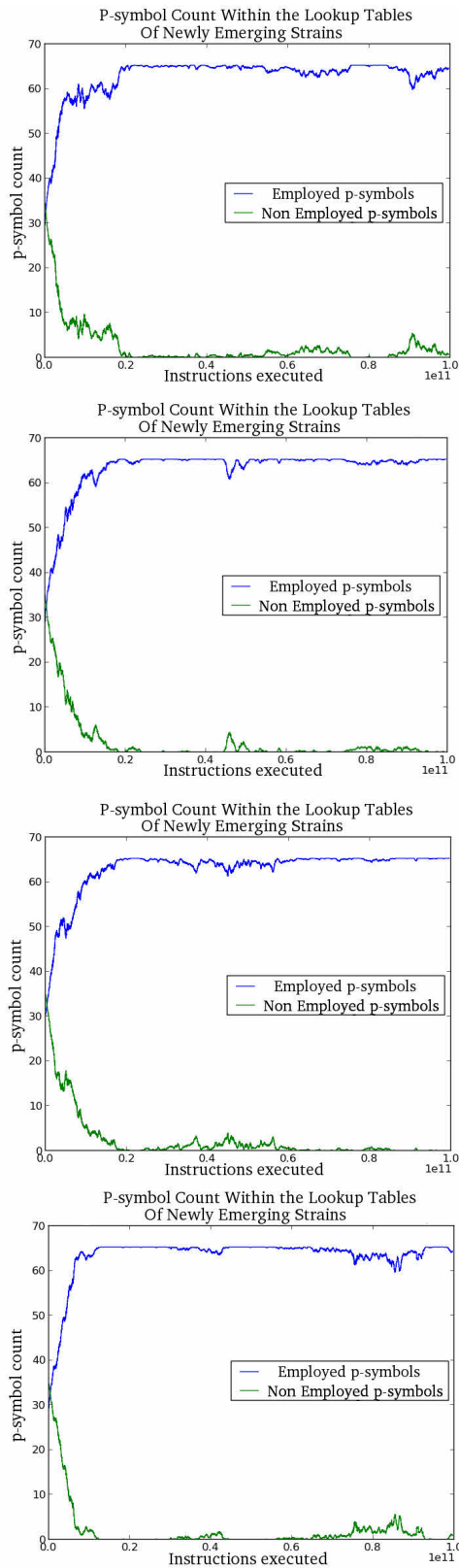


Figure 4: Four evolutionary simulations displaying the number of employed vs. non-employed p-symbols present in the lookup table of strains of newly emerging lineages.

generation ancestor’s lookup table and description. We can see here that the mapping is initially injective and surjective (bijective), as each element of the genotype space is mapped onto a different element of the phenotype space. This mapping is also invertible, as it is possible to determine the exact genotypic sequence by analysis of the phenotype alone. This can be denoted by saying that  $P \triangleq \phi(G)$  and  $G \triangleq \psi(P)$  where  $\psi() = \phi^{-1}()$ .

Figure 5(b) represents an offspring which experienced a perturbation to the third position in the lookup table’s description. For von Neumann reproduction, there is a generation delay between when a perturbation occurs in a genotype and when the perturbation is expressed in the phenotype. When this 2nd generation creature attempts to reproduce, it must first copy its exact genotype to the 3rd generation offspring Figure 5(c). The 2nd generation creature must then decode its own genotype, and construct the 3rd generation creature’s phenotype. However, under construction of the phenotype, when decoding the 2nd position<sup>10</sup> in the lookup table description, the employed p-symbol, `nop0`, is written to the third position in the lookup table, and the previous non-employed p-symbol, `nop2`, is lost from the genotype-phenotype mapping.

Even if the perturbed position in the lookup table description gets perturbed back to the previous state, Figure 5(d), the non-employed p-symbol does not return to the phenotype. This is because the genotype-phenotype mapping has been changed, and now the silent g-symbol, which initially was mapped onto a non-employed p-symbol, `nop2`, is now mapped onto an employed p-symbol, `nop0`.

We also see that the mapping is now non-injective and non-surjective, as an element of the phenotype space, `nop0`, is mapped onto more than one element of the genotype space, 0 and 2. Furthermore, an element of the phenotype space, `nop2`, is not mapped onto by any element of the genotype space. This renders the mapping invertible, as it is now impossible to determine the exact genotypic sequence via inspection of the phenotype alone as  $\psi() \neq \phi^{-1}()$  as now  $G = \psi^*(P)$ .

The only method in which a lost non-employed p-symbol can return to the lookup table is via a genotypic perturbation, which returns the lookup table description to its previous state, followed by a phenotypic perturbation, which directly introduces the lost non-employed p-symbol to the lookup table. Due to this level of ease at which a non-employed p-symbol can be lost, and the level of difficulty required to re-introduce the non-employed p-symbol to the mapping, there is a strong immediate bias present, which quickly eliminates all non-employed p-symbols from the lookup table.

This feature of the implemented mapping system demonstrates how *phenotypic* perturbations are inheritable under the circumstance that the perturbation affects the function

<sup>10</sup>Using zero-based indexing.

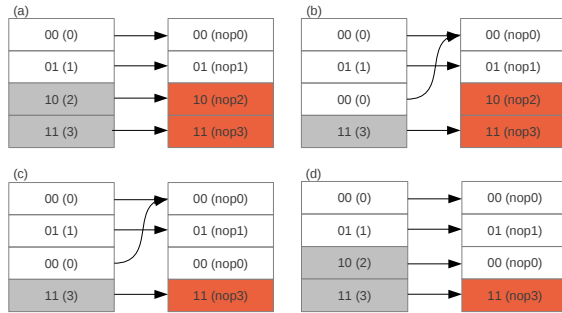


Figure 5: Perturbations to the lookup table description

$\psi()$ . If we have a machine  $X = (A' + B + C + D) + \phi(A + B + C + D)$ , where  $A'$  represents a general constructor with a changed genotype-phenotype mapping  $\psi^*$ , then this perturbation will be inherited to the offspring phenotype only if  $\psi^*(G') = \psi(G')$ , where  $G' = \phi(A' + B + C + D)$ . In other words, this demonstrates an instance of Lamarckian inheritance, where a perturbation of the phenotype is passed down to future generations without any change to the genotype. However, in order for this to occur, the perturbation must affect the component of the phenotype which decodes the genotype, such that the genotype will now be decoded to give rise to the new perturbed phenotype.

**Mutational robustness and Darwinian selection.** Darwinian selection may then sharpen the genotype-phenotype mapping, and create a more mutationally robust genotype. The allocation in which silent g-symbols are mapped upon employed p-symbols may be subject to Darwinian selection. Following a perturbation to a g-symbol, a phenotype may still preserve form if both g-symbols transcribes to the same p-symbol.

If there was no redundancy in the genotype space, then the mapping cannot incorporate inherent mechanism to ensure stability to perturbations and help the phenotype preserve its form under inheritable variation. Every p-symbol will have the same robustness to mutation, no matter how frequently its description occurs in the genotype, or how imperative it is to the correct operation of the phenotype.

For our experiments, we only have 28 employed p-symbols, but 64 g-symbols. Darwinian selection may select how the silent g-symbols are mapped upon the employed p-symbols. A p-symbol which is very common within the phenotype, has a high probability of having its description perturbed within the genotype. If a large percentage of the silent g-symbols are mapped upon the most frequent, employed p-symbols, then the phenotype will have an increased probability of holding form following an inheritable perturbation to the genotype. To test this hypothesis, a creature was

engineered with a non-surjective genotype-phenotype mapping. All silent g-symbols were mapped onto the employed p-symbol, `nop0`. `nop0` is very frequent throughout the phenotype, as it is used for template addressing. The mutational robustness of `nop0`'s description has now greatly increased, as there are 36 possible perturbations which will still allow the phenotype to preserve form. The Tierra soup was inoculated with two von Neumann self-reproducers, the original ancestor with a surjective genotype-phenotype mapping, and the engineered ancestor with the non-surjective genotype-phenotype mapping. The two creatures used different address templates, so that the descendants of each ancestor could be distinguished from each other. This simulation was run for 100 billion instructions and the experiment was repeated 100 times. It was found that in 76 instances the initial ancestor was driven to extinction, while in 24 instances the engineered ancestor with the non-surjective genotype-phenotype mapping was driven to extinction. These preliminary tests show that there may be a selective advantage for distributing the silent g-symbols amongst the most frequently occurring employed p-symbols, and therefore room for Darwinian selection to guide the evolution of the genotype-phenotype mapping. However, this work is pending and requires further experimentation.

The Tierra source code for these experiments along with the analysing software can be found at: [http://alife.rince.ie/evosym/alife\\_2013\\_dbbm.zip](http://alife.rince.ie/evosym/alife_2013_dbbm.zip).

**Acknowledgements.** This work has been supported by the European Complexity Network (Complexity-NET) through the Irish Research Council for Science and Technology (IRCSET) under the collaborative project EvoSym. Computing facilities were facilitated by The Irish Centre for High-End Computing (ICHEC).

## References

- Baugh, D. and McMullin, B. (2012). The emergence of pathological constructors when implementing the von neumann architecture for self-reproduction in tierra. In *From Animals to Animats 12*. Springer.
- Hasegawa, T. and McMullin, B. (2012). Degeneration of a von neumann self-reproducer into a self-copier within the avida world. In *From Animals to Animats 12*, pages 230–239. Springer.
- McMullin, B. (2000). John von neumann and the evolutionary growth of complexity: Looking backward, looking forward. *Artificial Life*, 6(4):347–361.
- McMullin, B. (2012). Architectures for self-reproduction: Abstractions, realisations and a research program. In *Artificial Life 13*, pages 83–90.
- Ray, T. (1991). An approach to the synthesis of life. *Artificial life II*, 10:371408.

- von Neumann, J. (1948). The general and logical theory of automata. In *Cerebral Mechanisms in Behaviour*, pages 1–32.
- von Neumann, J. (1966). *Theory of self-reproducing automata*. Edited and completed by A.W. Burks.

# Indirectly Encoded Sodarace for Artificial Life

Paul Szerlip and Kenneth O. Stanley

Department of EECS, University of Central Florida, Orlando, FL 32826-2363  
pszerlip@eecs.ucf.edu

## Abstract

The aim of this paper is to introduce a lightweight two-dimensional domain for evolving diverse and interesting artificial creatures. The hope is that this domain will fill a need for such an easily-accessible option for researchers who wish to focus more on the evolutionary dynamics of artificial life scenarios than on building simulators and creature encodings. The proposed domain is inspired by Sodarace, a construction set for two-dimensional creatures made of masses and springs. However, unlike the original Sodarace, the *indirectly encoded Sodarace* (IESoR) system introduced in this paper allows evolution to discover a wide range of complex and regular ambulating creature morphologies by encoding them with *compositional pattern producing networks* (CPPNs), which are an established indirect encoding originally introduced for encoding large-scale neural networks. The result, demonstrated through a technique called novelty search with local competition (which are combined through multiobjective search), is that IESoR can discover a wide breadth of interesting and functional creatures, suggesting its potential utility for future experiments in artificial life.

## Introduction

An important aim of artificial life is to uncover the conditions that yield interesting discoveries in evolutionary domains. For example, researchers studying *open-ended evolution* (Channon, 2001a,b; Maley, 1999; Ray, 1992; Standish, 2003; Yeager, 1994) seek to produce dynamics that yield a continual stream of novel and potentially more complex phenotypes. Other approaches to evolving lifelike creatures focus less on the evolutionary dynamics than on a particular property like morphology (Joachimczak and Wrobel, 2012), locomotion (Clune et al., 2011; Lehman and Stanley, 2011a), or both (Auerbach and Bongard, 2012; Bongard and Paul, 2000; Hornby and Pollack, 2002; Krcak, 2007; Lehman and Stanley, 2011b; Sims, 1994). The promise of such investigations is that they can potentially reveal key conditions that lead to the most compelling or natural results.

However, a significant obstacle to entering this research area is the lack of standardized artificial creature domains and genetic encodings. Indeed, in almost all such work

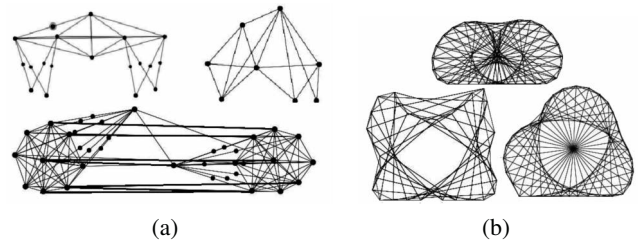


Figure 1: **Sodarace Examples.** Human-designed racers (a) exhibit diverse strategies and morphologies for ambulation while those produced through the evolutionary optimizer (b) share an amoeba-like morphology and similar ambulation.

researchers design their own domains and encodings from scratch, creating a high barrier to entry. In effect there is no out of the box domain that is easy to integrate quickly into a larger experimental framework. For example, an experiment aiming to research the impact of different evolutionary dynamics or selection pressures on open-ended discovery presently requires not only formulating a hypothesis, but also an entire domain and genetic encoding, which may not be the main motivation for the investigation in the first place. While of course sometimes researchers will prefer to build such experiments from the ground up, the availability of an easy, lightweight option that makes it possible to focus quickly on broader evolutionary questions would nevertheless be beneficial to the field overall.

The aim of this paper is to highlight such a lightweight option for a broad creature-space with low barrier to entry. The key concept is to introduce an encoding that opens a *Sodarace-like domain* of two-dimensional ambulatory creatures (McOwan and Burton, 2013, 2005) to broad and diverse evolutionary exploration and discovery. Sodarace is a simulation engine for two-dimensional creatures made of masses, springs, and muscles that ambulate based on the construction of their body morphology. It was originally introduced with human designers in mind, allowing them to construct their own Sodaracers by hand and then race them competitively, yielding a collection of diverse and interest-

ing human-designed morphologies reminiscent of the output of a successful artificial life world (figure 1a). In fact, one version of Sodarace even included an evolutionary optimizer, but because of the simplicity of its genetic encoding, the evolved morphologies only represent a small corner of the possibilities suggested by the human designs (figure 1b).

To open up such a domain to more interesting evolution, a Sodarace-inspired domain and encoding called *indirectly encoded SodaRace* (IESoR) is introduced in this paper that is designed explicitly for evolutionary exploration. In particular, the possibility of evolving a range of natural yet diverse morphologies exhibiting regularities across their structure is created by *compositional pattern producing networks* (CPPNs) (Stanley, 2007) evolved by the HyperNEAT algorithm (Gauci and Stanley, 2010; Stanley et al., 2009). While HyperNEAT and CPPNs were originally introduced to evolve large-scale neural networks, because the creatures in IESoR are also defined fundamentally by nodes and connections, HyperNEAT can in effect also evolve creature bodies with the same regularities and symmetries seen in CPPN-encoded neural networks (Clune et al., 2011).

With this new Sodarace-inspired implementation and an established indirect encoding behind it, the potential for the system to evolve both far more variety and quality than the original Sodarace evolver is demonstrated through a *novelty search with local competition* (Lehman and Stanley, 2011b), which is a recent method for efficiently surveying the range of possibilities that exist within a particular design space. The main outcome is that IESoR indeed introduces a rich and easily accessible platform for exploring a wide variety of interesting creatures with low simulation cost and concrete visual payoff.

## Background

This section contains an overview of Sodarace and MINS, the online projects that serve as the inspiration for IESoR, as well as a brief review of the NEAT and HyperNEAT methods used to encode creature morphology in IESoR.

### Sodarace

The Sodarace project is a simple two-dimensional physics world consisting entirely of masses, springs, and basic oscillatory muscles (McOwan and Burton, 2013, 2005). The goal in Sodarace is to create virtual robots and race them in different environments. Both the robots and the environments are usually hand-crafted by users. However, to aid in creating robots, a construction kit is provided to allow discovery and exploration by the community (McOwan and Burton, 2013, 2005).

The Sodarace project was originally conceived as a type of online Olympics meant to test humans against machine intelligence at the task of designing robot racers. In fact, one redesign of the software includes an evolutionary algorithm that optimizes morphologies for racing. Reflecting the

software's educational aspirations, an online repository of creatures and all relevant software packages are accessible in a centralized location (McOwan and Burton, 2013). At the peak of popularity, Sodaconstructor, the tool for creating the creatures, was played by about a million active users (McOwan and Burton, 2005), suggesting its potential as a platform for exploration and discovery.

### MINS

While Sodarace was a beacon for user creativity, the project itself was created over a decade ago and the community has declined since then. Nevertheless, the peak popularity of the project suggests the domain has wide appeal, though some aspects of the original software make Sodarace inaccessible to academic research. For example, as a closed source Java applet, certain parameters of the races cannot be modified because the Sodaconstructor user interface does not provide user access.

To address the obstacle to research, Stefan Westen created Mins Is Not Sodarace (MINS), an open source replica of the Sodarace environment (Westen, 2013). MINS is fully compatible with Sodarace, allowing the user to import design and environments from Sodaconstructor. By creating an alternative open-source environment, MINS allows adjusting parameters that are hardcoded inside the Sodarace domain. In an effort to curb cheating by Sodarace creatures, MINS alters the environment ceiling size, starting velocities, and maximum movement speeds.

MINS is an inspiration for the work in this paper in part because the variety of creature types found with the Sodaconstructor suggests the space of creatures is rich. MINS also shows that replicating the Sodarace environment is feasible and lightweight, while maintaining backwards compatibility. The primary principle extracted from Sodarace and MINS is the use of masses, springs, and muscles to construct varied creatures inside a customizable physics environment (Westen, 2013).

### NEAT and HyperNEAT

Evolving morphology and control is familiar to artificial life (Auerbach and Bongard, 2012; Bongard and Paul, 2000; Hornby and Pollack, 2002; Krcak, 2007; Lehman and Stanley, 2011b; Sims, 1994). In this spirit, recent additions to Sodarace include a utility with an evolutionary algorithm for creating and optimizing racers inside of the software (McOwan and Burton, 2005). In particular, inside of the Sodarace "Kiosk," users are presented with a limited interface for designing a creature. The user can adjust the number of virtual nodes and muscles along with the amplitude and frequency at which the muscles oscillate. The evolutionary algorithm then searches constrained by those parameters through selection and mutation to find the fastest racer possible. Inside of the Sodarace Kiosk, by default creatures appear to be circular in nature with criss-crossed inner connec-



tions. With gravity, the resulting creatures take on the shape of a semi-circular blob (figure 1b). The Sodarace community refers to this creature morphology as being “amoeba-like.” There is a Sodarace utility called the Amoebamatic that aids users in constructing these amoeba racers.

However, the most interesting handcrafted creatures from Sodarace generally do not exhibit amoeba-like properties. That is, the included genetic encoding is highly constrained to a small subset of all the interesting possibilities. Ideally, the encoding and evolutionary algorithm for evolving such racers would be able to search a wide breadth of possible creatures, which would make this kind of domain relevant to artificial life. Yet to efficiently search such a space requires a principled encoding capable of searching variable levels of complexity.

The first step towards this end in IESoR is the NeuroEvolution of Augmenting Topologies algorithm (NEAT) (Stanley and Miikkulainen, 2002, 2004). Although NEAT was originally introduced as a method for evolving artificial neural networks (ANNs), a major appeal of NEAT is its ability to evolve increasingly complex structures of any type, so that evolutionary search is not limited to a fixed space of possibilities.

Conveniently for IESoR, a method called Hypercube-based NEAT (HyperNEAT) (Gauci and Stanley, 2010; Stanley et al., 2009) builds on NEAT to help it encode large connectivity patterns with natural regularities like symmetry and repetition of structure. While such regularities are useful for neural networks, they can also in principle benefit *bodies* made of connections and joints in a similar way.

The key ingredient behind HyperNEAT is an indirect encoding called a *compositional pattern producing network* (CPPN) (Stanley, 2007). The idea behind CPPNs is that geometric patterns can be encoded by a composition of functions that are chosen to represent common regularities. The internal structure of a CPPN is a weighted network, similar to an ANN, that denotes which functions are composed and in what order, which means that instead of evolving ANNs as it normally does, NEAT can evolve CPPNs that generate a connectivity pattern across a network.

The difference in this paper is that the pattern encoded by CPPNs is interpreted as a body plan for a Sodarace-like creature instead of a neural network. In fact, in a significantly different domain, Auerbach and Bongard (2012) encoded the bodies of three-dimensional ambulating creatures with CPPNs. The indirect CPPN encoding can compactly represent patterns with regularities such as symmetry, repetition, and repetition with variation (Secretan et al., 2011; Stanley, 2007), which are also exhibited by many natural organism morphologies on Earth. In fact, part of the inspiration for CPPNs derives from observations of natural bodies (Stanley, 2007).

To understand how a composition of functions could represent these regularities, simply by including a Gaussian

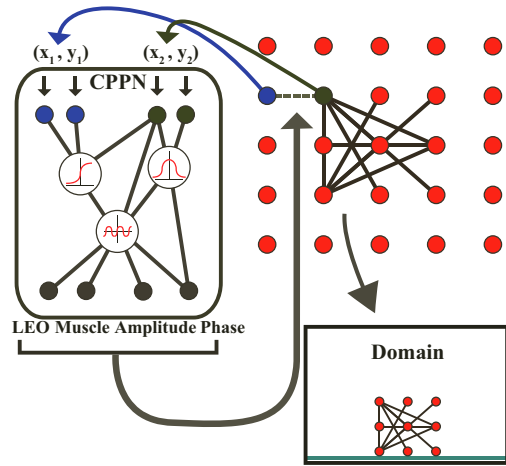


Figure 2: **Creating a Sodarace-like body using a HyperNEAT CPPN.** In regular HyperNEAT, the CPPN (left) would query the substrate (right) to determine the weights and presence (determined by the LEO output; Verbancsics and Stanley (2011)) of its connections. However, in IESoR the CPPN outputs the muscle, amplitude, and phase parameters for each queried connection instead of a connection weight. That way, the CPPN in effect describes the properties of a Sodarace body instead of a neural network, yet still with the same benefits of HyperNEAT as usual. The resultant creature is placed into a two-dimensional world where it attempts to ambulate.

function, which is symmetric, the output pattern of a CPPN can become symmetric. A periodic function such as sine creates segmentation through repetition. Most importantly, repetition with variation (e.g. such as the fingers of the human hand) is easily discovered by combining regular coordinate frames (e.g. sine and Gaussian) with irregular ones (e.g. the asymmetric x-axis). For example, a function that takes as input the sum of a symmetric function and an asymmetric function outputs a pattern with imperfect symmetry.

In this way, CPPNs produce regular patterns with subtle variations. The potential for CPPNs to represent patterns with motifs reminiscent of patterns in natural organisms has been demonstrated in several studies (Secretan et al., 2011; Stanley, 2007), and suggests such an encoding has potential in the domain of two-dimensional creatures.

Formally, CPPNs in HyperNEAT are functions of geometry (i.e. locations in space) that output connectivity patterns whose nodes are situated in  $n$  dimensions, where  $n$  is the number of dimensions in a Cartesian space. Consider a CPPN that takes four inputs labeled  $x_1$ ,  $y_1$ ,  $x_2$ , and  $y_2$ . This point in four-dimensional space also denotes the connection between the two-dimensional points  $(x_1, y_1)$  and  $(x_2, y_2)$ , and the output of the CPPN for that input thereby represents the weight of that connection (figure 2). By querying ev-

ery possible connection among a pre-chosen set of points in this manner, a CPPN can produce a connectivity pattern, wherein each queried point is a node position. Because the connections are produced by a function of their endpoints, the final structure is produced with knowledge of its geometry. In effect, the CPPN paints a pattern on the inside of a four-dimensional hypercube that is interpreted as the isomorphic connectivity pattern, which is the origin of the name hypercube-based NEAT (HyperNEAT). Connectivity patterns produced by a CPPN in this way are called substrates so that they can be verbally distinguished from the CPPN itself, which has its own topology. While the substrate in the original HyperNEAT is interpreted as an ANN, in IESoR the substrate is a creature's body, as explained next.

## Approach

This section describes the implementation details of IESoR, and explains the variant of HyperNEAT that enables it to create and evolve body plans.

### IESoR

IESoR implements three primary properties derived from Sodarace and MINS (figure 2):

1. The environment is two-dimensional and creatures consist solely of masses, springs, physical joints, and muscles.
2. In creature bodies, masses are implemented by nodes and springs are connections attached at the joints.
3. Muscles manipulate the length of connections, leading to motion.

In contrast to more complicated three-dimensional domains (Auerbach and Bongard, 2012; Krcah, 2007; Lehman and Stanley, 2011b; Sims, 1994), to support robust alife evolution IESoR is designed to be simple to modify and inexpensive to simulate. In the spirit of accessibility and extensibility of the Sodarace project, IESoR implements a Sodarace-like simulator in javascript built on top of Box2D (box2d.org), an open-source two-dimensional rigid body physics engine. There is a small performance hit for programming in a scripting language, but javascript allows the domain to be accessible through the browser for most modern computing devices, from phones to tablets to more traditional PCs. In addition, Box2D physics enables rich environments for testing creature morphologies. Finally, Box2D has been ported to most popular programming languages, which means IESoR could be ported without significant effort.

### Encoding Morphologies with HyperNEAT

Bodies inside of IESoR consist of masses with variable or fixed length constraints. Each constraint, or connection, is represented by a distance joint in Box2D (i.e. a constraint on the length between two masses) and has three distinct properties:

1. The joint is either variable or fixed length (i.e. a muscle or a bone).
2. The change in distance during muscle contraction is the muscle amplitude.
3. The phase shift of the sinusoidal function controlling muscle length is the muscle phase.

Fixed length connections, or bones, do not receive a magnitude or phase from the CPPN.

Recall that HyperNEAT paints a four-dimensional pattern across the weights of a network by querying a CPPN for every pair of nodes in the substrate. The insight in this paper is to take this concept of a substrate and extend it to two-dimensional morphologies. Instead of painting a pattern of weights across the substrate, the CPPN encodes both what joint constraints should exist between masses on a two-dimensional plane and their three virtual properties (i.e. bone or muscle, amplitude, and phase). For this purpose, the CPPN requires four outputs (as shown in figure 2).

Before clarifying how a HyperNEAT substrate can be used to represent a morphology, it is important to consider the placement of bones and muscles in natural body plans. The skeletal system is crucial to mobility at a fundamental level. Equally important to where bones are placed in a body plan is the concept of where bones *are not* placed. If a rough representation of the human body was drawn on a small grid of dots, the principle of symmetry is as important as the fact that there is no bone connecting the tip of the foot to the top of the skull. Morphologies generated in IESoR ideally also should usually respect this simple principle of locality.

Conveniently for this purpose, HyperNEAT can be expanded with a special Link Expression Output (LEO) (Verbancsics and Stanley, 2011) to generate an *expression pattern* that controls whether connections are expressed at different locations independently of other CPPN outputs. In Verbancsics and Stanley (2011), HyperNEAT with LEO was seeded with a bias towards favoring locality although evolution could adjust this bias during search.

To generate a morphology using an  $n$ -by- $n$  grid of nodes as the substrate, for each node location in the substrate (figure 2), the CPPN queries all other node positions. The  $(x, y)$  coordinate of nodes  $i$  and  $j$  are denoted as  $(x_i, y_i)$  and  $(x_j, y_j)$ , respectively. The input into the CPPN is thus  $x_i, y_i, x_j, y_j$ , and there are four outputs. First, the LEO output (which is a step function) is checked for a positive value. If LEO is positive, a connection is placed between nodes  $i$  and  $j$  from  $(x_i, y_i)$  to  $(x_j, y_j)$ . Then the output that determines whether the connection is a bone or a muscle is queried. If the output value is below a pre-defined *muscle threshold*, the connection becomes a fixed-length constraint. Otherwise, the constraint is a muscle, and the amplitude and phase of the muscle contraction are read from the remaining two CPPN outputs. Finally, to further reduce complexity in the resultant morphologies and keep computational costs

low, pairs of points greater than a third of the diagonal length of the substrate are not queried while constructing the two-dimensional creatures. An example of a fully constructed morphology is shown in the lower right of figure 2.

After assembling the masses and joints, the bodies are placed in a simple Box2D environment consisting of the ground, gravity, and friction. As the world is simulated, muscles oscillate according to the amplitude and phase values defined by the CPPN, while bones remain a fixed length. Creatures occupy distinct Box2D environments, and nodes cannot collide with each other.

## Experiment

Though its creatures are mainly hand-crafted, Sodarace shows that the space of possible two-dimensional body types is likely filled with creatures capable of movement. As noted in Section 2.3, the Sodarace Kiosk went on to create an automated approach to generating creatures, but resulted in a highly restricted space of bodies. The experiment described in this section is designed to show that not only is an automated approach capable of designing two-dimensional walkers, but the method can also produce a *wide variety* of different means for locomotion, thereby giving hope for further application of Sodarace-like creatures in artificial life.

### Novelty Search and Local Competition

To best demonstrate the morphological diversity possible in IESoR, Pareto multiobjective search (based on NSGA II) (Deb et al., 2002) including both novelty and local competition (Lehman and Stanley, 2011b) is implemented to explore the space of body types. Lehman and Stanley (2011b) first applied Pareto multi-objective search with novelty and local competition to yield a diverse group of ambulating three-dimensional morphologies all within a *single run* of evolution. Maintaining and exploiting diversity across evolution is both an impressive and important part of validating the potential for future artificial life research with IESoR.

The first of the three objectives that make up novelty search plus local competition is novelty search, which was introduced by Lehman and Stanley (2008, 2011a) to avoid the common pitfall of evolution prematurely converging on a deceptive objective. Novelty search aligns well with the aim of this experiment because the hope is to find a diversity of novel creatures. Joachimczak and Wrobel (2012) have shown before that novelty search can be effective for this purpose. The characterization of creature novelty for the novelty search component can significantly impact evolution and strongly bias the resulting creatures discovered. In this experiment, novelty search characterizes creatures by their width, height, and mass (as measured by the number of nodes and the sum of the connection lengths) at the first time-step of the simulation, which should lead to a visually diverse population. The novelty metric is the squared Euclidean distance separating two individuals in this charac-

terization space, and thus the novelty of a creature is proportional to how different its starting morphology is from that of other creatures currently in the population. Such a characterization space especially encourages creatures with varying widths, heights, and masses.

The second objective, local competition to walk farthest, forces individuals to compete *only* with those who are characterized as similar (Lehman and Stanley, 2011b). The idea is that within novelty search it is possible to push individuals who are similar with respect to the behavior characterization to compete locally to be the best of their type. That way, globally novelty search probes a wide variety of possibilities, but locally individuals optimize to be the best they can. In IESoR, creatures who are locally close share similar widths, heights, and masses, ideally indicating a similar morphology. Local competition is the mechanism for pressuring individuals with related morphologies towards more effective locomotion.

As in Lehman and Stanley (2011b), the Pareto multi-objective search has three objectives: novelty, local fitness, and finally *genotypic diversity*. The genotypic diversity objective encourages exploring innovative genotypes by assigning higher values to more novel genotypes. That way, new genotypes created by HyperNEAT are not initially penalized and thereby have a chance to optimize to reach their potential. This genetic diversity objective is in effect a multiobjective-compatible substitute for the usual speciation mechanism in NEAT, which serves the same purpose. Additionally, the genotypic diversity objective is also localized within the characterization space; similar in motivation to that of local competition, local genetic diversity ensures that genotypic diversity is not only exploited in those characterization niches in which such diversity is incidentally most easily expressed.

In all setups, the distribution of individuals in behavioral space as well as their overall performance is recorded. The idea is to quantify how much morphological diversity is discovered and maintained and how well each behavioral niche is being exploited overall throughout a run.

### Experimental Parameters

The overarching multiobjective algorithm is based on NSGA II (Deb et al., 2002). The population size is 120, and a run consists of 1,200 generations, resulting in 144,000 total genomes evaluated. The nearest-neighbor size for novelty search and local competition is 20. The three morphology dimensions used to characterize novelty (i.e. width, height, and mass) are rescaled so that their values fill the range between zero and three. The selection method for NSGA II was tournament selection (with tournament size two), and other parameters followed precedent Lehman and Stanley (2011b), which in turn used the parameters of Krcak (2007).

## Results

The intention of the experiment is to demonstrate that a wide variety of walkers exists in the encoding space defined by IESoR, thereby establishing the viability of IESoR for future alife research. Thus, as opposed to machine learning experiments aimed at demonstrating optimality, the aim in this experiment is to show both diversity and competence. Recall also that novelty search plus local competition is designed to return a significant coverage of possible solutions from a single long run. There is precedent for demonstrating the diversity that results from such a search. For example, Lehman and Stanley (2011b) measured the height and mass of three-dimensional morphologies from novelty search plus local competition to show the breadth of morphologies discovered by evolution, while Joachimczak and Wrobel (2012) used principal component analysis (PCA) to demonstrate coverage across morphological space after novelty search. Following this precedent, to quantify IESoR's ability to create diverse walkers, PCA is run across characterizations all 144,000 creatures from 1,200 generations of evolution to create a visualization of the resultant diversity.

In particular, to characterize morphological diversity in IESoR for the purpose of visualization, three dimensions that describe gross creature characteristics (i.e. width, height, and mass) are projected into a two-dimensional space by the PCA algorithm. However, while PCA with this information can reveal the diversity across the morphological space, the goal of this analysis is also to give a sense of the competence of such creatures as well. That way it becomes possible to observe the *diversity of competent creatures* instead of just diversity overall. Therefore, in the visualization of the PCA output in figure 3, to ensure the graph shows the diversity of only competent walkers, only points for walkers that ambulate beyond 200 units (which is several times a creature's maximum body length) are displayed. Furthermore, the size of each point's radius is proportional to the absolute distance traveled by the creature beyond 200 units.

Because thousands of points result, the visualization is further refined to reduce clutter and ensure that each point represents a genuinely unique individual. For this purpose, the plane is discretized into  $40 \times 40$  equally sized "bins." Creatures are placed into bins according to the coordinate assigned by the PCA process. Conceptually, each bin thus represents a similar area in morphological space, and the creature assigned to a bin that traveled the farthest among all in that bin is chosen as the representative of that bin. That way, the circles in figure 3 show the best performance for the morphological class represented by its respective bin, and each circle represents a distinct morphological class. Any bin without a representative (shown as empty space in figure 3 lacks an individual who could ambulate at least 200 units.

Of the 1,600 possible bins, 450 are filled with individuals who can ambulate the minimum distance, covering in total 28.1% of all possible bins. Furthermore, the visualization

in figure 3 exhibits the breadth of coverage of competent morphologies. In effect, IESoR with novelty search plus local competition uncovered hundreds of unique and effective ambulation methods covering a significant breadth of conceivable strategies.

Equally important as this quantitative perspective is a qualitative analysis of the breadth of behaviors. It is important to note that every behavior in figure 3 can be viewed at <http://eplex.cs.ucf.edu/ecal13/demo/PCA.html> through a special online interface where the user can click on any point and see the corresponding creature behavior. This fast interactive visualization of hundreds of creatures is possible in part due to the lightweight, inexpensive nature of Sodarace-like creatures, which is one of their potential advantages for researchers in artificial life. Figure 3 also shows a sampling of morphologies, while figure 4 shows a subset of those at different stages of ambulation.

An additional important further qualitative observation is the significantly broader diversity seen in IESoR compared to the original Sodarace evolver's amoeba-like creatures shown in figure 1b. Among those that can be observed are gaits based on loping (degrading into pushing) (figures 3a/4a), pogo-stick hopping (3b/4b), multiple cascading octopus legs (3c/4c), dragging (3d/4d), complex galloping (3e), sliding and pumping (3f), and bouncing into a long dive (3g). Some strategies depend on an initial burst of propulsion, while others rely on stable and consistent ambulation. Some of the very best gaits (largest circles in figure 3) involve galloping or hopping, though even among the very best the diversity of approaches is significant.

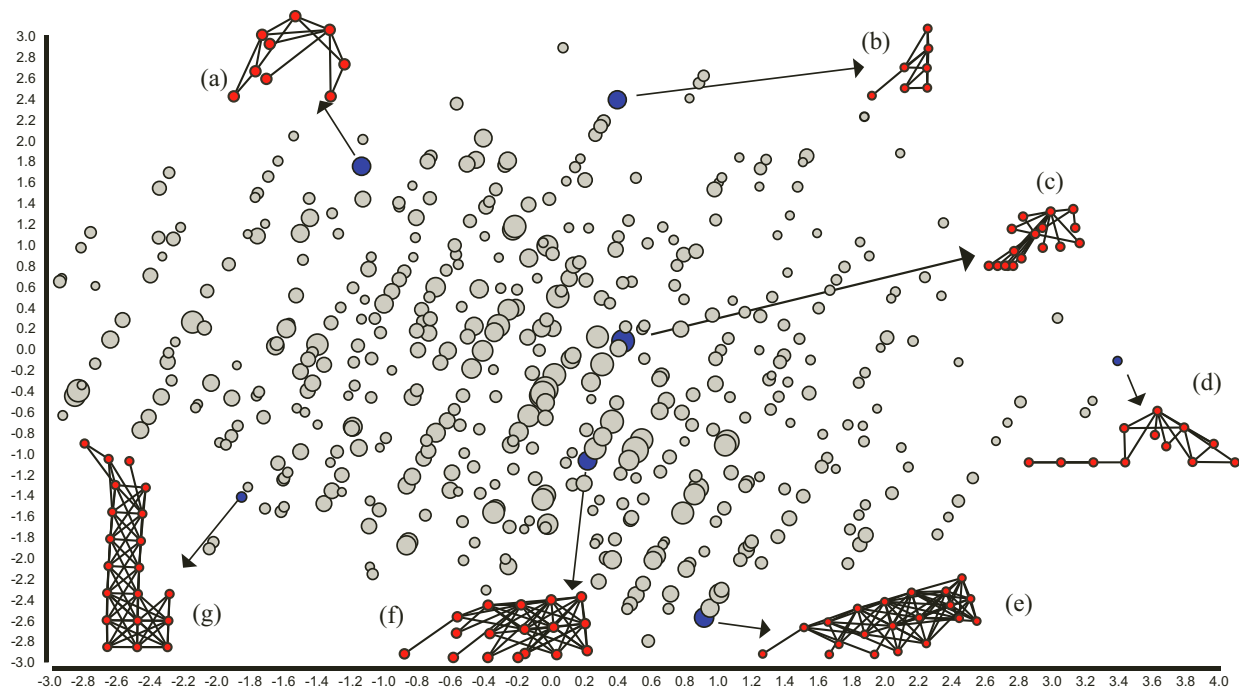
## Discussion

The most important implication of the results is that Sodarace-like domains do contain a diversity of viable walkers that can be systematically discovered with the right encoding and selective pressures. Unlike with the early evolver application built for Sodarace, morphologies evolved in IESoR do not exhibit only a single stereotypical organizational motif. Instead, they ambulate in many different ways, from legged-style locomotion to serpentine pulsation to periodic lunges, suggesting the potential for more elaborate applications of this kind of technology in the future.

For example, because the IESoR creatures are inexpensive to simulate, they are amenable to web-based interaction. In fact, the demonstration at the website actually simulates evolved discoveries in real-time through javascript, obviating the need even for video. This ease of simulation means that IESoR creatures can smoothly integrate into interactive evolutionary applications, artificial life worlds, or even interactive simulations allowing human intervention. Additionally, the framework can potentially be extended to three dimensional creatures using current browser technologies.

Investigations relevant to artificial life on open-ended evolution (Channon, 2001b; Maley, 1999; Ray, 1992;





**Figure 3: PCA-based Visualization of Morphological Diversity and Performance.** The location of each point represents its respective creature morphology, while the size indicates the absolute fitness. All points shown are for creatures able to walk at least 200 units. A total of 28.1% of all 1,600 possible bins are filled with competent walkers, suggesting the diversity of ambulation methods. Furthermore, several creatures are shown to give a sense of qualitative diversity. The creatures for every point in this visualization can also be viewed in motion at <http://eplex.cs.ucf.edu/ecal13/demo/PCA.html>.

Standish, 2003; Yaeger, 1994) and diversity maintenance (Lehman and Stanley, 2011b; Mouret and Doncieux, 2012) can thus be quickly set up and conducted in the future with IESoR. Creatures can also potentially move beyond forward ambulation to more complex interactions such as foraging or predation. To facilitate such future applications, code for IESoR is available at <https://github.com/OptimusLime/IESoR>.

## Conclusion

The paper demonstrated IESoR, a lightweight two-dimensional platform for evolving ambulating creatures inspired by Sodarace (McOwan and Burton, 2013, 2005). The aim is to provide an accessible platform to artificial life researchers that is inexpensive to simulate. That way, artificial life experiments that previously required significant up-front design can become easier to ramp up and build quickly. Results from searching through the indirectly-encoded creature space in IESoR with novelty search plus local competition suggest that the space indeed contains a breadth of feasible morphological discoveries with functional ambulatory capabilities, suggesting that IESoR is potentially a viable platform for artificial life research in the future.

## Acknowledgements

This work was supported by the National Science Foundation under grant no. IIS-1002507 and also by a NSF Graduate Research Fellowship. Special thanks to Peter W. McOwan and Edward J. Burton for creating the original Sodarace, and to Stefan Westen for creating Mins.

## References

- Auerbach, J. E. and Bongard, J. C. (2012). On the relationship between environmental and morphological complexity in evolved robots. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2012)*, pages 521–528, New York, NY. ACM Press.
- Bongard, J. C. and Paul, C. (2000). Investigating morphological symmetry and locomotive efficiency using virtual embodied evolution. In *Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, pages 420–429. MIT Press.
- Channon, A. (2001a). *Evolutionary Emergence: The Struggle for Existence in Artificial Biota*. PhD thesis, University of Southampton.
- Channon, A. (2001b). Passing the alife test: Activity statistics classify evolution in geb as unbounded. In *Proceedings of the European Conference on Artificial Life (ECAL-2001)*. Springer.
- Clune, J., Stanley, K. O., Pennock, R. T., and Ofria, C. (2011). On the performance of indirect encoding across the continuum of regularity. *IEEE Transactions on Evolutionary Computation*.



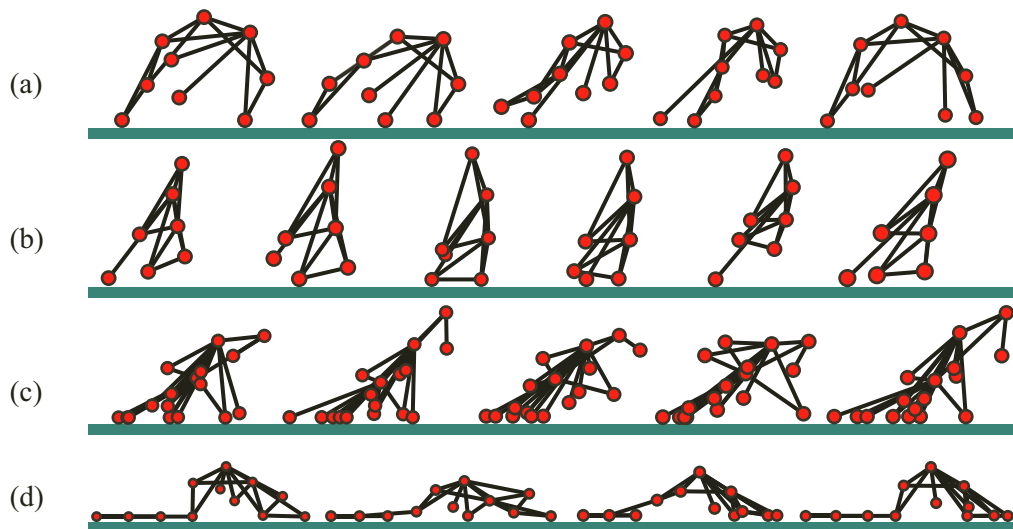


Figure 4: **Creature Motion Over Time.** The motion (from left to right over time) of a small sample of successful creatures evolved in IESoR is shown. The letters (a)–(d) correspond to those in figure 3.

- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.
- Gauci, J. and Stanley, K. O. (2010). Autonomous evolution of topographic regularities in artificial neural networks. *Neural Computation*, 22(7):1860–1898.
- Hornby, G. S. and Pollack, J. B. (2002). Creating high-level components with a generative representation for body-brain evolution. *Artificial Life*, 8(3).
- Joachimczak, M. and Wrobel, B. (2012). Open ended evolution of 3D multicellular development controlled by gene regulatory networks. In *Proceedings of the Thirteenth International Conference on Artificial Life (ALIFE XIII)*, pages 67–74, Cambridge, MA. MIT Press.
- Krcak, P. (2007). Evolving virtual creatures revisited. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007)*, New York, NY, USA. ACM Press.
- Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In Bullock, S., Noble, J., Watson, R., and Bedau, M., editors, *Proceedings of the Eleventh International Conference on Artificial Life (Alife XI)*, Cambridge, MA. MIT Press.
- Lehman, J. and Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223.
- Lehman, J. and Stanley, K. O. (2011b). Evolving a diversity of virtual creatures through novelty search and local competition. In *GECCO '11: Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218, Dublin, Ireland. ACM.
- Maley, C. C. (1999). Four steps toward open-ended evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-1999)*, volume 2, pages 1336–1343, Orlando, Florida, USA. IEEE Press.
- McOwan, P. and Burton, E. (2000–2013). Sodarace website. [URL http://sodarace.net](http://sodarace.net).
- McOwan, P. W. and Burton, E. J. (2005). Sodarace: Adventures in artificial life. In *Artificial Life Models in Software*, pages 97–111. Springer.
- Mouret, J.-B. and Doncieux, S. (2012). Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evolutionary computation*, 20(1):91–133.
- Ray, T. (1992). Evolution, ecology and optimization of digital organisms. Technical Report Working paper 92-08-042, Santa Fe Institute.
- Secretan, J., Beato, N., D'Ambrosio, D. B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J. T., and Stanley, K. O. (2011). Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3):345–371.
- Sims, K. (1994). Evolving 3D morphology and behavior by competition. pages 28–39. MIT Press, Cambridge, MA.
- Standish, R. (2003). Open-ended artificial evolution. *International Journal of Computational Intelligence and Applications*, 3(167).
- Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines Special Issue on Developmental Systems*, 8(2):131–162.
- Stanley, K. O., D'Ambrosio, D. B., and Gauci, J. (2009). A hypercube-based indirect encoding for evolving large-scale neural networks. *Artificial Life*, 15. To appear.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127.
- Stanley, K. O. and Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100.
- Verbancsics, P. and Stanley, K. O. (2011). Constraining connectivity to encourage modularity in HyperNEAT. In *GECCO '11: Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 1483–1490, Dublin, Ireland. ACM.
- Westen, S. (2004–2013). Mins website. [URL http://mins.sourceforge.net/](http://mins.sourceforge.net/).
- Yaeger, L. (1994). Computational genetics, physiology, metabolism, neural systems, learning, vision and behavior or polyworld: Life in a new context. In Langton, C. G., editor, *Artificial Life III, Proceedings Volume XVII*, pages 263–298. Addison-Wesley.

# An Energy-Based Model for Spatial Social Networks

Alberto Antonioni<sup>1</sup>, Mattia Egloff<sup>2</sup>, Marco Tomassini<sup>1</sup>

<sup>1</sup>Information Systems Department, Faculty of Business and Economics,

<sup>2</sup>Section of Computer Science and Mathematical Methods, Faculty of Arts and Humanities,  
University of Lausanne, Switzerland

## Abstract

In the past decade, thanks to abundant data and adequate software tools, complex networks have been thoroughly investigated in many disciplines. Most of this work has dealt with networks in which distances do not have physical meaning and are just dimensionless quantities measured in terms of edge hops. However, in many cases the physical space in which networks are embedded and the actual distances between nodes are important, such as in geographical and transportation networks. The Random Geometric Graph (RGG) is a standard spatial network model that plays a role for spatial networks similar to the one played by the Erdős–Rényi random graph for relational ones. In this work we present an extension of the RGG construction to define a new model to build bi-dimensional spatial networks based on energy as realistic constraint to create the links. The constructed networks have several properties in common with those of actual social networks.

## Introduction

Social networks arise in a wide range of contexts and are really pervasive in our society. Examples range from corporate partnership connections, scientific collaborations, sexual contacts, film actors networks, to Facebook and other online social networks among others. In recent years much attention has been given to model these networks in order to gain a better understanding of their general structures and their functions like information flow, locating individuals, disease spread, etc. There is an increase in the number of network models in the literature (Toivonen et al., 2009) but, although general features common to all social networks are reproduced, such as the typically high clustering, none of them can represent all the typical characteristics of social networks in a realistic way. This is due to the fact that all these networks have formed and grow in ways that are similar but not identical. In other words, each actual network is an instance of a class of possible realizations and its particular structure depends on its particular history, frozen structures, dynamics, and many other factors.

It is generally believed that social networks possess the following main features:

- *positively skewed degree distribution*: the majority of agents have relatively small degrees, while a small number of agents may have large degrees.
- *high average clustering coefficient  $\bar{C}$* : the conditional probability that two neighbors of an agent will be connected is much higher than what would be expected in a sparse random graph.
- *positive degree correlations*: the degrees of the neighbors are not independent and are similar on average.
- *small average shortest path length  $\bar{L}$* :  $\bar{L} \propto \log(N)$  i.e. it is rather small compared to the network size  $N$ .
- *existence of community structure*: clusters of agents which are highly connected within themselves but loosely connected to other subgroups.

In the last decade social networks as a topology-free relational graph structure have been much studied. A comprehensive literature would be too long to mention, but review articles and books contain a wealth of information on them e.g. (Boccaletti et al., 2006; Newman, 2010). Relational networks are those in which actual distances do not count and path lengths are computed by simply adding one for each link in the path. Social networks such as coauthorship networks are usually taken to be relational. However, Euclidean space is an important factor in many networks, for example transportation and communication networks among others are of this kind (see (Barthélemy, 2011) for a recent good review of the field). But spatial aspects can also be important for social networks. For example, while two Facebook friends might live in the US and Europe respectively and still be represented by a standard link in the net, it is also observed that many links will be among people that are geographically close. Thus, spatial considerations may play a role in social networks too. Therefore, while purely relational social network models are important and have been studied in depth (see e.g. Toivonen et al. (2006); Vázquez (2003); Kumpula et al. (2007); Catanzaro et al. (2004)), their spatial aspects are much less known (but see the following

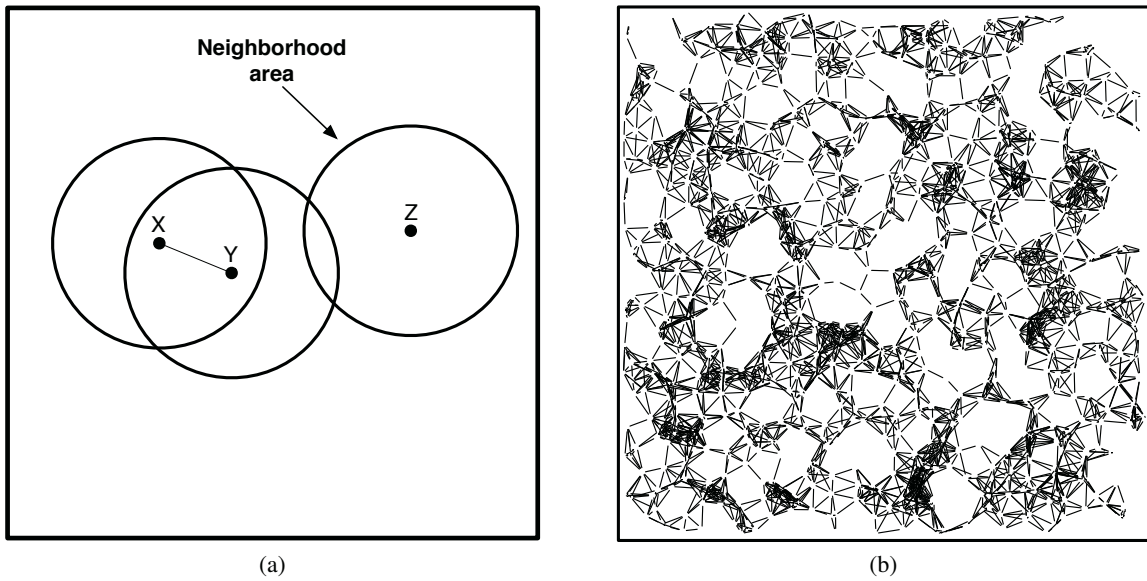


Figure 1: (a): Neighborhood areas of nodes  $X$ ,  $Y$  and  $Z$ . (b): An example of RGG with  $N = 1000$  and  $R = 0.056$ , and average degree  $\bar{k} = 10$  (for illustrative purposes the unit square is bounded).

works Boguñá et al. (2004), Wong et al. (2006) and Serrano et al. (2008) for some recent attempts). Our model is very simple and it is intended to be only a first step toward more realistic ones. It is based on the concept that each agent is initially given a constant amount of energy to be used to establish links with other agents. The spatial bias is given by the fact that links cost less if they are made with agents that are physically closer. The model gives rise to networks that have high clustering coefficient, positive degree assortativity, and modularity due to the appearance of communities. The degree distribution is rather peaked but the model could be easily modified to produce broader distributions.

The article has the following organization. In the next section we give a brief introduction to random geometric graphs, a spatial model that will be needed in the sequel. Next we describe our own model of a social spatial network. The following section presents and discusses the main numerical results, and we then give our conclusions.

### Random Geometric Graph

The Random Geometric Graph (RGG) is obtained when the points located in the plane are connected according to a given geometric rule. The simplest rule is a proximity rule which states that nodes only within a certain distance are connected. There is an extensive mathematical literature on geometric graph and the random case was studied by physicists in the context of continuum percolation (Penrose, 2003; Dall and Christensen, 2002; Barthélemy, 2011).

In this work we refer to the following construction process for a RGG with  $N$  nodes and radius  $R$ :

- the  $N$  nodes are placed on the unit space  $\Omega \in \mathbb{R}^2$  with

uniform distribution,

- an edge is created for every pair of nodes whose distance is  $r < R$ . The distance is given by the standard Euclidean metric on  $\mathbb{R}^2$ .

Furthermore, we shall assume that the unit space  $\Omega$  is the square  $[0, 1]^2$  with cyclic boundary conditions (torus). In Fig. 1a three nodes  $X$ ,  $Y$ ,  $Z$  and their neighborhood areas are depicted. Nodes  $X$  and  $Y$  are connected through an edge since they are within the neighborhood area of each other respectively, while  $Z$  is not connected to  $Y$  even if it is sharing a common area with it. Figure 1b shows a RGG realization with  $N = 1000$  and  $R = 0.056$ ; in this case, for illustrative purposes,  $\Omega$  is a bounded unit square.

It is also possible to adopt different shapes of neighborhood area generated according to other metrics. For example, the Manhattan distance is sometimes used to model mobility networks (Glauche et al., 2003; Di Crescenzo et al., 2012). The general properties of these networks are very close to those using the more common Euclidean distance, which are the ones we describe here.

The average degree  $\bar{k}$  of a RGG can be easily estimated by the formula  $\bar{k} = \rho V$ , where  $\rho$  is the node density, representing the number of nodes within a unit space, and  $V$  is the neighborhood area. In this case  $\rho = N$ , since  $\Omega$  is an unit space, and  $V = \pi R^2$ . In conclusion,  $\bar{k} = \pi N R^2$ .

The degree distribution of RGGs with a sufficiently large number of nodes can be estimated by the Poisson distribution with parameter  $\lambda = \bar{k}$  (Dall and Christensen, 2002).

The average clustering coefficient is given averaging on all node's individual clustering coefficients (Newman,

2010). This property on RGGs was extensively studied in the work of Dall and Christensen (2002), in which they have found the law for the average clustering coefficient as a function of the dimension of the space. Here the dimension is equal to two, and it is possible to demonstrate that the average clustering coefficient tends to  $1 - \frac{3\sqrt{3}}{4\pi} \sim 0.5865$ , for large values of  $N$  and for all 2-dimensional RGGs in the Euclidean space. This important result depends on the particular construction of RGGs. The average clustering coefficient tends to the ratio of the average shared neighborhood area of two connected nodes and the whole neighborhood area. It is clear that changing the radius  $R$  this fraction maintains the same value.

Due to its construction process, in a RGG there is positive degree-degree correlation. This property is commonly detected studying the *assortativity coefficient*, which is the Pearson correlation coefficient of degree between pairs of connected nodes (Boccaletti et al., 2006). In the recent work of Antonioni and Tomassini (2012), it has been demonstrated that the assortativity coefficient tends to the average clustering coefficient value for any  $d$ -dimensional RGG. Many other properties of RGGs have been studied in Penrose (2003).

### Energetic Spatial Network

In order to construct more realistic social networks with spatial structure, we now consider the following two realistic assumptions for a spatial social network (see Fig. 2):

- *limited neighborhood*: a given node may create links only within the set of nodes in its neighborhood area given by the radius  $R$ .
- *distance cost*: creating a long link is more costly in terms of energy expenditure than creating one closer to the focal node.

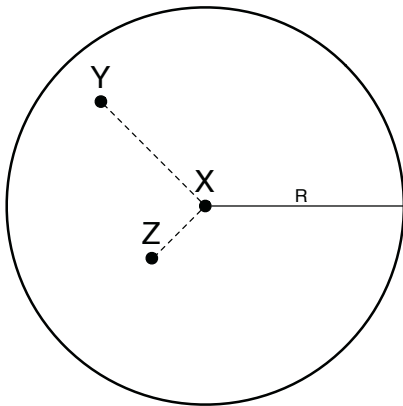


Figure 2: Node  $X$  can be linked only to nodes within its neighborhood area given by radius  $R$ . In this case, creating a link with  $Y$  is more costly than creating one with  $Z$ .

In this model we assume that *energy*, which is constant and the same for each node, is a resource provided to nodes in order to create and maintain their links. We shall call networks constructed according to this model as Energetic Spatial Networks (ESNs). Our hypothesis is thus that each node has limited energy available to create its acquaintances, which are increasingly costly with increasing physical distance. This feature can be also assumed in actual social networks, in which real distance may play an important role in order to establish a connection. It is indeed reasonable to think that in order to minimize the efforts and to maintain a social tie most individuals tend to be connected with their spatial neighbors, at least in social networks that are not fully mediated by communication devices. The present model creates a static network; dynamical aspects might be included in the future by requiring that maintaining links through time also costs a certain amount of energy.

The construction process to build ESNs with  $N$  nodes, radius  $R$ , and initial energy  $E$ , can be summarized as follows:

1. The  $N$  nodes are randomly placed with uniform distribution on the unit space  $\Omega \in \mathbb{R}^2$ . All nodes have the same initial energy equal to  $E$ .
2. A node  $X$  is picked uniformly at random in the set of all nodes, and  $Y$  is chosen uniformly at random in the set of nodes whose distance from  $X$  is  $r < R$ .
3. An edge between  $X$  and  $Y$  is created if  $d_{XY}$ , which is the Euclidean distance between  $X$  and  $Y$ , is less than  $E_X$  (residual energy of  $X$ ) and  $E_Y$ . If the edge is created, then the residual energies of  $X$  and  $Y$  are both decremented by  $d_{XY}$ .
4. Steps 2–3 are repeated until no more edges can be created according to the linking rule.

The unit space  $\Omega$  can be seen, similarly to the RGGs construction, as the square  $[0, 1]^2$  with cyclic boundary conditions (torus). It is rather clear that this construction process produces RGGs for  $E \rightarrow \infty$ , while, for  $\{R, E\} \rightarrow \infty$  complete graphs are obtained.

### Results

Figure 3 shows empirical values of the normalized average degree, average clustering coefficient, and assortativity coefficient of a realization of an ESN as a function of the initial energy  $E$  (Fig. 3a), and of the radius  $R$  (Fig. 3b). The normalized average degree  $k_{norm}$  shown in Figs. 3 is obtained by dividing the average degree  $\bar{k}$  of an ESN by the average degree of a RGG with the same radius  $R$ . This means that for  $\bar{k}_{norm} \rightarrow 1$  the ESN can be approximated by a RGG. In Fig. 3a, the parameters of an ESN are  $N = 10000$  and  $R = 0.04$ . We can observe that for high values of the initial energy  $E$  ( $> 1.5$ ), ESN features become closer to those of a



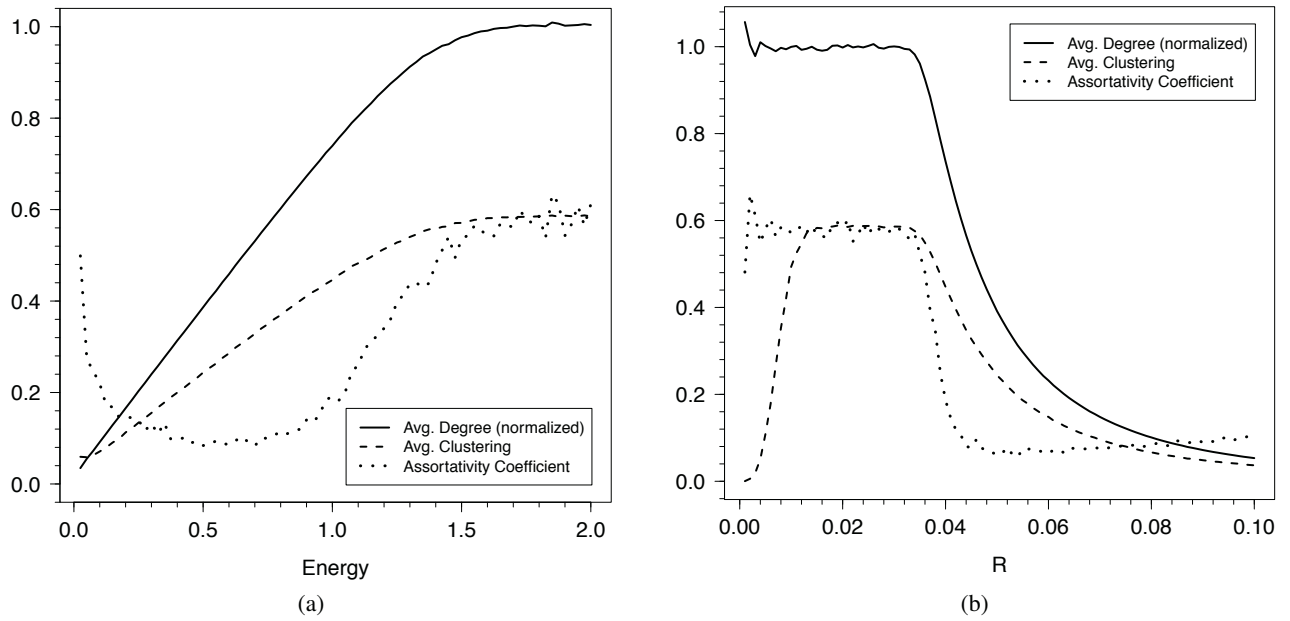


Figure 3: Numerical values of normalized average degree, average clustering coefficient, and assortativity coefficient of an ESN as a function of : (a) the initial energy  $E$  and (b) the radius  $R$ . (a): The other ESN parameters are  $N = 10000$  and  $R = 0.04$ . The average degree is normalized dividing by 50, which represents the average degree value of a RGG with parameters  $N = 10000$  and  $R = 0.04$ . (b): The other ESN parameters are  $N = 10000$  and  $E = 1.0$ . The average degree value is normalized dividing by  $10000\pi R^2$ , which represents the average degree of a RGG with  $N = 10000$ .

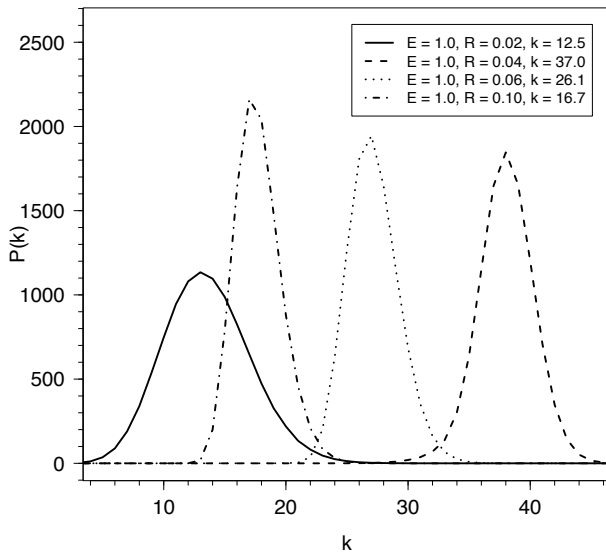


Figure 4: Degree distribution functions for ESNs with  $N = 10000$  and  $E = 1.0$ . The other parameter, the radius  $R$ , assumes the following values: 0.02, 0.04, 0.06 and 0.1.

RGG since in this case the energy is more than enough to allow the creation of all possible links within radius  $R$ . In fact, average clustering and assortativity coefficients both tend to the characteristic RGG value 0.5865. Therefore, with this model, it is possible to select a value for  $E$  that will produce a desired high clustering coefficient for the ESN. This is not possible with the standard RGG model, which converges by construction to a fixed value.

In Fig. 3b, the parameters of the ESN are  $N = 10000$  and  $E = 1$ . For small values of  $R$  ( $< 0.04$ ), ESNs can be considered as RGGs, because the radius is rather small compared to the initial energy and the nodes can build all the possible links in their neighborhood areas. For larger values of  $R$  the network becomes more interesting. The clustering coefficient and the assortativity coefficient both decrease and thus suitable values of  $R$  for a social network are around 0.04 – 0.06. The normalized average degree is approximately 1 for  $R < 0.04$ , but then tends to zero for larger values of  $R$ . This means that ESNs become sparser than a RGG with the same radius  $R$  and thus  $R$  should not go beyond 0.06 for a realistically connected network.

For  $E = 1$ , Figure 4 depicts the degree distribution functions of four realizations of ESNs with different values of  $R$ . The thick curve with  $R = 0.02$  gives rise to a standard RGG as we have seen above (see also Fig. 3b). The other curves correspond to three ESNs and are rather peaked. Although relational social networks usually have rather broad



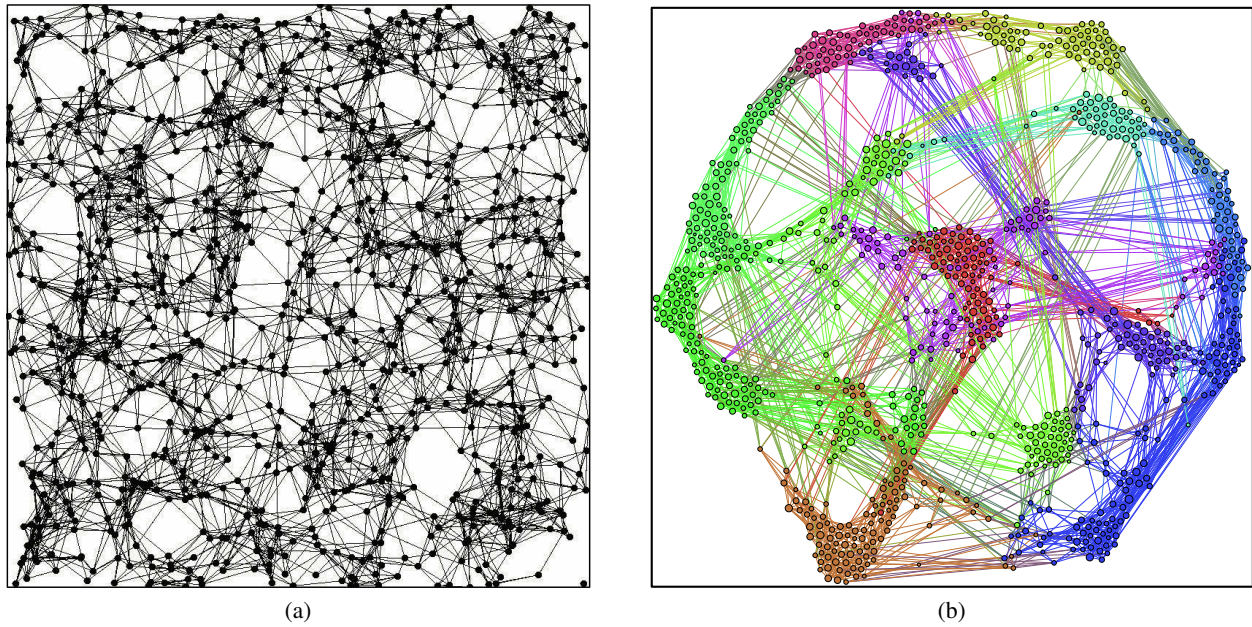


Figure 5: Realization of an ESN with  $N = 1000$ ,  $E = 0.64$ , and  $R = 0.1$ . The average degree is about  $\bar{k} = 10$ , while the average clustering coefficient is equal to 0.205 and the assortativity coefficient is 0.135. The modularity is  $M = 0.758$  (Blondel et al., 2008). (a) Spatial representation, and (b) graphical representation given by the OpenOrd algorithm (Martin et al., 2011).

degree distribution functions (Newman, 2001a,b; Barabási et al., 2002; Tomassini and Luthi, 2007), spatial constraints do not allow agents to have too many links to other nodes. The effect is particularly evident in technological and transportation networks where hard constraints such as rail crossing or economic factors in cable length in power grids, for instance, set well defined limits to the network's connectivity (see Barthélemy (2011) and references therein). The above factors arising from spatial physical constraints are less important for social acquaintances but, to some extent, they also influence the spatial structure of social networks. Indeed, social networks can actually be seen as a mix of relational and spatial factors. However, part of the observed effect on the degree distribution function is certainly attributable to the fact that we give the same constant amount of initial energy to all nodes. A simple improvement of the model would consist in attributing energy according to a more complex distribution such as a power-law or another suitable function.

The presence of communities, which are clusters of densely connected nodes, is a common feature of all social networks (Newman, 2010). Communities may arise in many ways, for instance people having common interests, people sharing the same culture or religion, going to the same school, living in the same area, as is the case in our model networks, and many others. We have studied the cluster structure of ESNs using one of the several heuristic community detection algorithms (Blondel et al., 2008)

and we have found that ESNs present several communities. Figure 5 shows an ESN with  $N = 1000$ ,  $E = 0.64$ , and  $R = 0.1$ . Figure 5a is a true geographical representation of the network and, although it might superficially look very similar to the RGG image of Fig. 1b, it possesses features such as the presence of longer links and the lack of linking to all the neighbors falling into the disk-shaped neighborhood area for a given node. Instead, Fig. 5b shows with different colors the communities found in the network by the community detection algorithm. To improve the rendering, Fig. 5b does not take into account the Euclidean metrics in the node and link disposition. Modularity  $M$  (Newman, 2006) for the graph is rather high at a value of  $M = 0.758$  which means that the communities are rather well defined. Although the  $M$  measure is not without drawbacks, it still provides a rather clear-cut indication of the significance of the network's community structure.

## Conclusion

In this article we have proposed an original model for the construction of social networks having a spatial dimension. We started from the random geometric graph model and we added a few ingredients in order to generate networks that possess most of the statistical features shown by actual spatial social networks. The main idea is to attribute a limited amount of energy to nodes, the same for all of them. Nodes can spend this resource to link to other nodes as a function of their Euclidean distance, longer links being more expensive

than shorter ones. In this way we obtain networks that do indeed look similar in many ways to actual ones from the point of view of their statistical features. In particular, the modeled networks have high clustering, positive degree correlation, and the presence of community structure. Moreover, properties such as clustering and assortativity can be tuned to some extent by changing the parameters  $E$  and  $R$ . The degree distribution function is peaked due, in part, to spatial constraints but especially because of the homogeneous distribution of energy and radii among the nodes. This consideration suggests ideas for further research with the purpose of getting more realistic social networks. For example, one could assume that nodes are not placed at random on the space, in order to form even more clustered networks and have more community structures. Furthermore, it is reasonable to consider non-constant distributions of the energy among the nodes which would probably produce some more connected individuals and thus a broader degree distribution. The linking process is bilateral in the present version, i.e. both partners must pay the same amount of energy to create the connection. One-way links could also be considered and the model could be extended to make it dynamical allowing for link suppression as well as link formation.

### Acknowledgments

The authors thank François Bavaud for carefully reading the manuscript and for his useful comments.

### References

- Antonioni, A. and Tomassini, M. (2012). Degree correlations in random geometric graphs. *Physical Review E*, 86:037101.
- Barabási, A.-L., Jeong, H., Nédá, Z., Ravasz, E., Schubert, A., and Vicsek, T. (2002). Evolution of the social network of scientific collaborations. *Physica A*, 311:590–614.
- Barthélemy (2011). Spatial networks. *Physics Reports*, 499:1–101.
- Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:P10008.
- Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U. (2006). Complex networks: Structure and dynamics. *Physics Reports*, 424:175–308.
- Boguñá, M., Pastor-Satorras, R., Díaz-Guilera, A., and Arenas, A. (2004). Models of social networks based on social distance attachment. *Physical Review E*, 70:056122.
- Catanzaro, M., Caldarelli, G., and Pietronero, L. (2004). Assortative model for social networks. *Physical Review E*, 70(3):037101.
- Dall, J. and Christensen, M. (2002). Random geometric graphs. *Physical Review E*, 66:016121.
- Di Crescenzo, G., Kondareddy, Y., and Zhang, T. (2012). Concrete synthetic modeling of vehicular networks as random geometric graphs. In *Communications (ICC), 2012 IEEE International Conference on*, pages 1–5. IEEE.
- Glauche, I., Krause, W., Sollacher, R., and Greiner, M. (2003). Continuum percolation of wireless ad hoc communication networks. *Physica A: Statistical Mechanics and its Applications*, 325:577–600.
- Kumpula, J. M., Onnela, J.-P., Saramäki, J., Kaski, K., and Kertész, J. (2007). Emergence of communities in weighted networks. *Physical Review Letters*, 99:228701.
- Martin, S., Brown, W. M., Klavans, R., and Boyack, K. W. (2011). Openord: an open-source toolbox for large graph layout. In *IS&T/SPIE Electronic Imaging*, pages 786806–786806. International Society for Optics and Photonics.
- Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.
- Newman, M. E. J. (2001a). Scientific collaboration networks. I. network construction and fundamental results. *Physical Review E*, 64:016131.
- Newman, M. E. J. (2001b). Scientific collaboration networks. II. shortest paths, weighted networks, and centrality. *Physical Review E*, 64:016132.
- Newman, M. E. J. (2010). *Networks: An Introduction*. Oxford University Press, Oxford, UK.
- Penrose, M. (2003). *Random Geometric Graphs*. Oxford University Press, Oxford, UK.
- Serrano, M. A., Krioukov, D., and Boguná, M. (2008). Self-similarity of complex networks and hidden metric spaces. *Physical Review Letters*, 100:078701.
- Toivonen, R., Kovanen, L., Kivelä, M., Onnela, J.-P., Saramäki, J., and Kaski, K. (2009). A comparative study of social network models: Network evolution models and nodal attribute models. *Social Networks*, 31:240–254.
- Toivonen, R., Onnela, J.-P., Saramäki, J., Hyvönen, J., and Kaski, K. (2006). A model for social networks. *Physica A: Statistical Mechanics and its Applications*, 371:851–860.
- Tomassini, M. and Luthi, L. (2007). Empirical analysis of the evolution of a scientific collaboration network. *Physica A*, 385:750–764.
- Vázquez, A. (2003). Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E*, 67:056104.
- Wong, L. H., Pattison, P., and Robins, G. (2006). A spatial model for social networks. *Physica A: Statistical Mechanics and its Applications*, 360(1):99–120.

# Analysis of Path Planning Algorithms : a Formal Verification-based Approach

Arash Khabbaz Saberi<sup>1</sup>, Jan Friso Groote<sup>1</sup> and Sarmen Keshishzadeh<sup>1</sup>

<sup>1</sup>Eindhoven University of Technology, Eindhoven, The Netherlands  
a.khabbaz.saberi@student.tue.nl, j.f.groote@tue.nl, s.keshishzadeh@tue.nl

## Abstract

With the emergence of multi-robot systems in the field of robotics there is a need for new approaches for modeling and investigating the behavior of robotic systems. Formal verification is a well-known mathematical method which has been used for decades in order to expose potential design faults in industrial systems. In this paper we introduce the application of formal verification techniques in the context of multi-robot systems. Applying verification techniques, we aim to prove that the collective behavior of a group of robots satisfies certain desired properties. We illustrate our approach using a simple path planning algorithm which conducts a set of robots from their initial positions to their destinations on a planar surface.

## Introduction

Multi-robot systems have the potential to solve complicated problems by their collective behavior. In recent years multi-robot systems have attracted the attention of many researchers. The challenge is to control the collective behavior of a number of homogeneous and simple robots in order to accomplish sophisticated tasks. Robustness and versatility are the main characteristics offered by multi-robot systems compared with single robot systems.

The path planning problem for multi-robot systems is of great importance. The objective is to find, or construct a trajectory for each robot to start moving from its initial position toward its destination while avoiding collisions. Path planning for mobile robots has been mainly addressed in the literature from two different aspects. The first approach is to construct trajectories for robots and move them along the predefined trajectories (Laumond, 1998; Luigi Biagiotti, 2009; Saska et al., 2006). The second approach is the algorithmic approach, in which robots use an algorithm to find a way toward their destinations (Jain et al., 2010; min Han, 2007; Pamosoaji and Hong, 2011). Furthermore, existing analysis approaches for robotic systems are mostly focused on the behavioral analysis of individual robots. However, with the increasing popularity of multi-robot systems new approaches are required to analyze the collective behavior of these systems.

In the context of the algorithmic approach to path planning, we propose a new method for behavioral analysis of multi-robot systems. In this approach, a well-known formal verification technique called model checking (Baier and Katoen, 2008) is applied to formally analyze the behavior of a robotic system. Model checking is an algorithmic approach which verifies the validity of a desired system property against a high-level model of the system. Model checking algorithms exhaustively explore all possible behaviors specified by the system model and check whether this model meets the given requirement. This technique is now applied to industrial software/hardware systems in order to verify the correctness of their behavior against a set of requirements. In this paper, we apply it for the analysis of a multi-robot system. The advantage of our approach is that it provides a systematic methodology for modeling and analyzing the collective behavior of a group of robots.

Fig. 1 depicts an overview of our approach. Given an informal specification of a path planning algorithm and the way robots realize this algorithm, we specify a formal model of the robotic system as a set of communicating processes in the mCRL2 language (Groote et al., 2009). Moreover, desired properties of the system are formalized in a mathematical language. In particular, we use the modal  $\mu$ -calculus (Groote and Mateescu, 1999) for property specification. The verification procedure verifies the formalized property against the system model. If the property is satisfied by the model, the verification procedure will respond with a “Yes”. Otherwise, a “No” response is returned. In this case, a counterexample is also generated which describes a situation that the property is violated by the model. We apply the mCRL2 toolset (Cranen et al., 2013) for verification of our case study.

*Related work* The application of formal methods in the context of robotic systems has been studied in different research works. Most of these works focus on constructing paths in a robot’s workspace that satisfy properties specified in a mathematical language. Given an initial position of a robot in its workspace and a requirement specified in linear temporal logic (LTL), (Fainekos et al., 2005) apply a



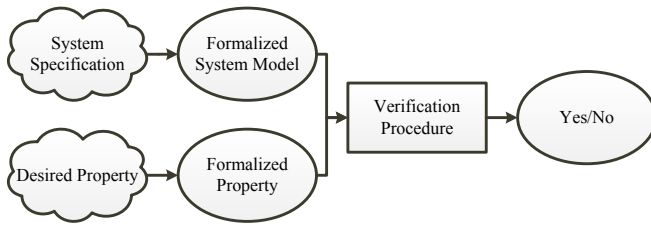


Figure 1: Overview of the Approach

discretization technique to generate a continuous trajectory that satisfies the LTL formula. In this method model checking tools are used for constructing trajectories. The work in (Kress-Gazit et al., 2009) describes an approach for generating a robot controller that guarantees the satisfaction of a given LTL property in all execution scenarios in certain execution environments. It is assumed that the behavior of robot's environment is specified as another LTL formula.

(Fainekos, 2011) proposes a diagnostic approach. First, they define a notion of closeness for properties specified in LTL. For an LTL property violated by the specification of the system they aim to find minimal changes to the formula that makes it satisfiable.

In (Quottrup et al., 2004) a high-level description of a timed automata-based approach for modeling and verification of a multi-robot system is provided. Due to a high level of abstraction only movements of robots are considered in the analyses. Moreover, they do not analyze concrete path planning algorithms. They apply this high level approach to evaluate certain characteristics of a robotic system (e.g., shortest time required by a robot moving in arbitrary directions to arrive at its destination) and construct paths to a destination using model checking evidence.

However, in this paper we include information about the sensing mechanisms of robots and their movements in our analysis. Hence, compared to the timed automata-based approach we investigate a lower level of abstraction. Our approach also offers a high-level language for specifying robotic systems. Moreover, the main objective of this approach is to prove that certain properties are valid/falsified in a system and provide efficient feedback to the designer of a robotic system.

**Overview** In section *System Specification* we specify the main characteristics of the robotic system that we consider as a case study in this paper. In section *Validation Properties* we discuss the types of properties that we aim to validate. Main features of our analysis approach are explained in section *mCRL2 & the Modal  $\mu$ -calculus*. We apply this approach in sections *Modeling a Multi-Robot System* and *Verification* to model and verify our case study. *Conclusion* includes some concluding remarks and suggestions for future research.

## System Specification

We introduce a simple multi-robot system which we use as a case study throughout the paper. In particular, the physical environment in which robots perform their tasks, main features of the robots (regarding their moving and sensing abilities), and the algorithm that guides robots movements are discussed in this section.

**Workspace** We assume that robots move on an unbounded planar surface. There are no static obstacles on the surface and robots are the only dynamic obstacles in this setting.

**Robots** Robot platforms designed for multi-robot applications such as (Bristeau et al., 2011; Mondada et al., 2003) assume that robots need limited computational resources to process their path planning algorithm. On the other hand, these robots are equipped with advanced hardware, e.g., sensors, and processors, to communicate with their environment or perform complicated tasks, e.g., transport heavy objects.

In our case study we consider an algorithm that can be realized by robots using limited resources. We assume that robots are capable of performing translation movements to move toward their destinations. Each robot can also perform in-place rotation movements (e.g., when facing an obstacle) in order to change its direction of movement. Moreover, each robot is only aware of the position that it currently occupies and the location of its destination. Every robot has an embedded sensing device which is capable of detecting obstacles in its range of sense.

**Algorithm** A robot starts from an initial position and tries to move toward its destination while actively scanning its range of sense for potential obstacles. Whenever a robot senses the presence of an obstacle in its sensing range, it performs an in-place 90-degree counter-clockwise rotation and scans its surroundings along the new direction. This mechanism is repeated until a safe direction is found.

It should be noted that the reactions of the robots are not affected by the actual positions of obstacles. When a robot finds a safe direction, it moves along that direction, steers itself to the destination, and repeats the path finding procedure until it arrives at its destination. In our analyses we do not make any assumptions about the relative processing rate of the robots. In other words, robots can perform their tasks with different processing rates. The same approach can be adapted to systems consisting of robots with identical processing rates.

## Validation Properties

In this section we describe several kinds of desired properties for a path planning algorithm. We aim to check for absence of deadlock, absence of collision, and reachability of robots to their destinations for a given path planning algorithm.

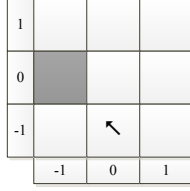


Figure 2: Sensing range for a specific position and movement direction

In order to perform such checks in a robotic system it is required to include information about timing, details of movement trajectories, accuracy of sensors, and shapes of the robots. To make our analyses feasible, we try to keep these details to a minimum. In our analyses we consider a specific abstraction of the robots workspace and robots behavior introduced in *System Specification*.

We assume that the robots workspace is a two dimensional grid which is uniformly partitioned into disjoint cells. Thus each cell in this setting can be characterized by a pair  $(x, y)$ . Assuming a specific position as the origin of the grid, for each robot the workspace is the following:

$$W = \{(x, y) | x, y \in \mathbb{Z}\}$$

Robots are modeled as objects without any specific shape. Each robot occupies a specific cell on the grid at each moment in time. We assume that robots can perform translation movements along vectors from the following set:

$$D = \{(i, j) | i, j \in \{-1, 0, 1\} \wedge (i, j) \neq (0, 0)\}$$

Translation movements to a neighboring cell are assumed to be atomic events without any time duration. Moreover, performing an in-place rotation by a robot does not change the configuration of the system.

We abstract from the complexities of the sensing mechanism and the way robots realize it. We assume that sensing is an atomic action. Performing this action, robots check for presence of obstacles in their range of sense. The sensing range is assumed to be one cell ahead from a robot's current position along the robot's next movement direction. In Fig. 2 the gray cell depicts the sensing range for a robot in the cell  $(0, -1)$  which intends to move along  $(-1, 1)$ . Sensors are assumed to be accurate and provide correct information.

Finally, we assume that sensing and moving are the only actions robots perform to realize a path planning algorithm. These actions are performed by robots in an infinite loop. Robots that arrive at their destinations can still sense their surroundings. In what follows we explain several properties that can be useful in the context of a path planning algorithm.

**Deadlock-freeness** Checking for absence of deadlock is one of the general checks that can be performed. Performing this check we can detect problematic situations in the

system where no further action can be performed by robots. As mentioned earlier, in our sketched system robots can always sense their surroundings. Thus, in this case deadlock freedom easily follows from the system specification. However, this check could be useful in general.

**Collision-freeness** For a given path planning algorithm, it is desirable to check that for all trajectories calculated by the algorithm, robots will never collide with an obstacle (which could be another robot). Considering our abstraction of the workspace and robots movements, this means that robots should not share a cell on the grid with another object.

**Reachability** The ultimate goal of a path planning algorithm is to guide robots to their destinations. We check whether robots can reach their desired destinations in a finite number of steps. It is also possible to define a limit on the number of steps that a robot takes before reaching its goal.

As each robot can prevent other robots from reaching their destinations, we are also investigating reachability while reaching the goal can be prevented by other robots for infinitely many times.

## mCRL2 & the Modal $\mu$ -calculus

In this section we explain the details of our analysis approach depicted in Fig. 1. In section *Labeled Transition Systems* we describe labeled transition systems as a means for modeling and analyzing the behavior of systems. In section *The mCRL2 language* we introduce mCRL2 as a high-level language for specifying labeled transition systems in a compositional manner. We use this language to capture the behavior of multi-robot systems by specifying and composing the behavior of its components. The modal  $\mu$ -calculus is the property specification language used in our approach which is briefly explained in section *The modal  $\mu$ -calculus*.

### Labeled Transition Systems

Modeling and analyzing the behavior of systems with labeled transition systems (LTS) is a common technique. A labeled transition system is a directed graph with a set of states, i.e., graph nodes, and a set of transitions, i.e., graph edges. States of an LTS correspond to system states. Transitions are labeled by actions and show the evolution of the system when executing a specific action. Labeled transition systems have an initial state which is depicted by an incoming arrow.

Fig. 3 is a simple LTS which specifies the following behavior for a system. Starting in the initial state an  $a$  action is performed. Then the system non-deterministically chooses to perform an  $a$  or  $b$ . Performing an  $a$  will result in a state where no further action can be performed by the system. Performing a  $b$  means that the left branch is executed. After the  $b$  action is done the system will perform  $a$  forever.



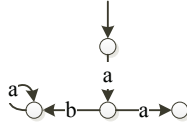


Figure 3: A simple labeled transition system

## The mCRL2 language

Modeling systems behavior with LTSs does not scale to complex systems. High-level languages can be used to describe complex LTSs. We use the mCRL2 language for this purpose. In this language LTSs are described as processes. Data specification and process specification are the main aspects of the language.

**Data specification** The data specification subset of the language allows the user to use the built-in data types (e.g., booleans, natural numbers) or to define and manipulate her own system-specific data types. For instance, in our case study we need two data types to specify the position of each robot and movement vectors. In the mCRL2 syntax we have:

```
sort Pnt = struct P(X:Int,Y:Int);
Dir = struct D(I:Int,J:Int);
```

 (1)

Each variable of type Pnt corresponds to a cell on the grid. Instances of Pnt can be constructed by applying the constructor P. For an instance p of type Pnt the first and second element of the pair can be accessed through the application of the projection functions X and Y, respectively. The same argument is applicable to Dir. In our modeling we always use instances d of Dir such that  $I(d), J(d) \in \{-1, 0, 1\}$ .

Useful operations on data types can be defined as functions. Each function is specified by its name, types of arguments and return type, and a set of equations describing the relation between the input(s) and output of the function. We illustrate this with an example:

```
map NextPos:Pnt # Dir -> Pnt;
var p:Pnt, d:Dir;
eqn NextPos(p,d)= P(X(p)+I(d), Y(p)+J(d));
```

 (2)

Given the current position of a robot and its next movement direction, NextPos computes its next position. In this example, we have one equation in our equation system (preceded by eqn). The var block preceding the equation system defines the variables used in the equation. These variables are used for pattern matching. The single equation specifies that for a given position p and direction d, the return value is an instance of Pnt. The X attribute of the next position is the sum of X(p) and I(d). The Y attribute is computed in a similar way.

**Process specification** The process specification aspect of the language provides a compositional way for system behavior specification. Actions are the main building blocks of processes. We assume that actions are atomic events without any time duration. Actions can also carry data parameters. For the specification of a robotic system, we can define an action which mimics a single step movement of a robot. The parameter of this action specifies the movement vector.

```
act move:Dir;
```

In order to achieve complex behaviors or communication patterns we can combine actions sequentially, in parallel, or make a non-deterministic choice between a (possibly infinite) set of actions. We can also let data values affect the behavior of a process by adding conditional expressions to processes.

The non-deterministic choice between processes  $P$  and  $Q$  is denoted by  $P + Q$ . The process  $P + Q$  will behave as  $P$  or  $Q$ . The sequential composition of  $P$  and  $Q$  is denoted by  $P.Q$  and the resulting process first performs the behavior of  $P$  and then behaves as  $Q$ . The notation  $P||Q$  represents the parallel execution of  $P$  and  $Q$ . It is also possible to enforce communication between actions executed by different processes. Using this facility together with the operator  $||$ , one can force communication between  $P$  and  $Q$  for specific actions. The rest of the actions will be interleaved. Assuming that  $c$  is a boolean expression, the process  $c \rightarrow P \diamond Q$  will behave as  $P$  if  $c$  is satisfied and will behave as  $Q$  otherwise.

Using the mCRL2 language, we can specify the movements of a robot as a process. For example, a robot that takes a sequence of non-deterministically chosen right and left steps can be specified as follows:

```
proc Robot(rp:Pnt) =
  move(D(1,0)).Robot(NextPos(rp,D(1,0))) +
  move(D(-1,0)).Robot(NextPos(rp,D(-1,0)));
```

The process carries a data parameter which represents the current position of Robot. After each movement along  $D(1,0)$  or  $D(-1,0)$  the robot behaves as the Robot process with a new parameter calculated by NextPos.

One can easily influence the choices of Robot by another process, e.g., Environment. In our context this process can mimic the behavior of the environment by collecting information about the workspace and enforcing certain constraints on movements. We can establish a communication between the two processes to enforce a movement in a specific direction. To realize this communication in this example, Environment should perform an action, e.g., en\_move, which carries data parameters of the same type as move. We describe Environment as follows:

```
proc Environment =
  en_move(D(1,0)).en_move(D(-1,0)).Environment;
```

By enforcing the communication between move and en\_move and executing Robot and Environment in parallel, the required controlling mechanism will be achieved. In this way,

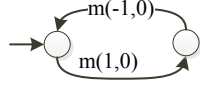


Figure 4: LTS for Robot | Environment with enforced communication

*move* and *en\_move* are only executed in a synchronized manner both carrying equal data parameters. In our simplified example Environment only affects the movement direction of Robot by enforcing it to move to its right and left alternately. Putting Robot and Environment in parallel will result in the LTS of Fig. 4 where action *m* represents the communication of *move* and *en\_move*.

### The modal $\mu$ -calculus

We use the modal  $\mu$ -calculus for specifying system properties. The following grammar gives the basic form of modal  $\mu$ -calculus formulae where *a* is an action.

$$\begin{aligned} \phi ::= & \text{true} \mid \text{false} \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \\ & \langle a \rangle \phi \mid [a] \phi \mid \mu X. \phi \mid \nu X. \phi \end{aligned} \quad (3)$$

The operators  $\wedge$ ,  $\vee$ , and  $\neg$  have their usual meanings. The formula  $\langle a \rangle \phi$  is valid in a state of an LTS when an action *a* can be performed such that  $\phi$  is valid after this action *a* has been done. The formula  $[a] \phi$  is valid in a state if all possible executions of action *a* lead to a state where  $\phi$  holds.

The formulae  $\mu X. \phi$  and  $\nu X. \phi$  are the minimal and maximal fixed points, respectively. In both cases, *X* usually occurs in  $\phi$ . The property  $\mu X. \phi$  is valid for all the states in the smallest set *X* that satisfies ' $X = \phi_s$ '. Similarly,  $\nu X. \phi$  is valid for all the states in the largest set *X* that satisfies ' $X = \phi_s$ ' where  $\phi_s$  represents all states where  $\phi$  is valid. Other capital letters (e.g., *Y*) can also be used instead of *X*.

As an example consider a setting with parameterless actions  $\{\text{move}, \text{sense}\}$ . The property  $\mu X. ([\text{sense}]X)$  is valid if *sense* is executed for a finite number of times. In other words, *move* is unavoidable (unless a deadlock occurs). The property  $\nu X \mu Y. ([\text{move}]X \vee [\text{sense}]Y)$  specifies that any subsequence of consecutive *sense* actions should be finite.

It is often useful to use a variant of the grammar in Eqn. 3 which allows the occurrence of multiple actions or a sequence of actions in a modality. For this purpose regular formulas are used within modalities. The formula *true* represents the set of all actions and  $\cup$  (union),  $\cap$  (intersection), and  $\neg$  (complement with respect to the set of all actions) can be used to specify action formulas. Regular formulas extend action formulas to allow the use of sequences of actions in modalities. For instance for a subset of actions  $\alpha$ ,  $\alpha^*$  denotes any sequence of actions from  $\alpha$ .

Since actions can carry data parameters, we need ways to refer to data values in formulae. As an example consider the

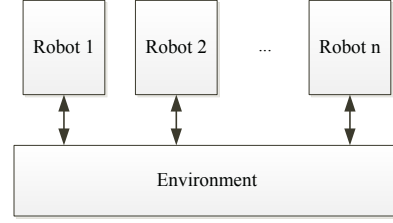


Figure 5: Robot and Environment processes

set of actions  $\{\text{move}, \text{sense}\}$  both carrying one data parameter of type *Dir*. Data can be introduced to modal formulae referring to *move* and *sense* using existential or universal quantifiers. Quantification can be used within modalities. For instance  $\forall d : \text{Dir} : (\neg \text{move}(d))^* \phi$  says that as long as there is no move in any direction,  $\phi$  should hold. Quantifiers can also be used outside the scope of modalities with their standard meaning. For instance  $\forall d : \text{Dir}. [\text{move}(d)] \phi$  is true if  $\phi$  holds after performing *move* in any direction. We can also store and process data values in fixed points. Using this feature it is possible for instance to specify constraints on the number of certain events. Consider the following formula:

$$\mu X(p : \text{Pnt} = P0). (\forall d : \text{Dir} [\text{move}(d)] X (\text{NextPos}(p, d)) \vee (X(p) > Y(p)))$$

Here *p* records the current position of the robot and is initially set to the *P0* (the initial position). The property says that after finite number of movements the robot should be in a certain part of the grid where  $X(p) > Y(p)$  holds.

We refer the interested reader to (Groote and Mateescu, 1999) for a more detailed explanation of the modal  $\mu$ -calculus and its semantics.

### Modeling a Multi-Robot System

In this section we elaborate on the simple modeling scheme we introduced in the previous section to formalize the multi-robot system described in section *System Specification*. Fig. 5 depicts a schematic view of our modeling approach. For a system consisting of *n* robots we specify *n* + 1 processes, i.e., *n* robot processes and 1 environment process, in mCRL2 and put them in parallel. This scheme conforms to our description in *System Specification*, i.e., robots communicate with the environment but they do not have direct communication among themselves.

Each process carries and manipulates certain data parameters. Every robot process carries parameters which indicate its current position and the (potential) direction of the next move. We assign a unique identifier to each robot process so the environment process can distinguish them when performing communications. The environment process carries data parameters to record the current position of all the robots. Since the destinations of the robots are not affected

by the dynamics of the system we model them as global parameters. For instance for Robot1 from Fig. 5 we can specify the destination as the position (2, 3) as follows:

```
map PD1:Pnt;
eqn X(PD1)=2;Y(PD1)=3; (4)
```

In our case study we assume that the system consists of identical robots. Thus, the processes we use to describe their behavior only differ in the unique parameter that is used for the identification of the robots. This approach can be adapted to systems consisting of robots with nonidentical path planning algorithms. For the sake of simplicity we study a system with two robots in this paper. The following definition specifies a data type with two unique instances which we use for robots identification.

```
sort ID = struct id1|id2;
```

In what follows we explain the mCRL2 processes that we use to describe the robots and the environment.

**Robots** In our abstraction of the robotic system, each robot scans its surroundings and performs movements in an infinite loop. Thus, robot processes can be specified in terms of actions *rs* (sense) and *rm* (move).

We enforce a communication on *rs* with the environment to collect information about the presence of obstacles. A robot performs *rs* providing its unique identifier and the next movement direction. Performing *rs* a robot should be able to react to both outcomes of the performed check, i.e. presence or absence of obstacles. We specify *rs* as follows:

```
act rs:ID # Dir # Bool; (5)
```

Scanning the range of sense along vector *d* by a robot identified by *id1* can be modeled as a nondeterministic choice between a “Yes” or “No” response for presence of obstacles, i.e., *rs*(*id1*,*d*,*true*)+*rs*(*id1*,*d*,*false*).

To establish a communication on *rs*, the environment process should perform an action, e.g., *es*, with identical parameter values. Given the movement direction of the robot, if the environment does not find an obstacle in the range of sense it will perform *es*(*id1*,*d*,*false*). Next, the robot will perform a move action along the direction vector. Otherwise, the environment will perform *es*(*id1*,*d*,*true*) and the robot should not move.

A robot identified by *id1* performs *rm*(*id1*,*d*) to declare its movement along the vector *d*. We enforce a communication between the robots and the environment on this action. In this way the environment can calculate the new position of the moving robot and update its information. The following process describes the behavior of a robot identified by *id1*. The process Robot2 can be specified in a similar way.

```
proc Robot1(p:Pnt,d:Dir) =
sum b:bool. rs (id1,d,b).b ->
  Robot1(p,NextDir(id1,d,p,b)) <>
  rm(id1,d).Robot1(NextPos(p,d),
    NextDir(id1,d,NextPos(p,d),b));
```

The syntax **sum** *b*: **bool**. *rs* (*id1*,*d*,*b*) is a shorthand for *rs*(*id1*,*d*,*true*)+*rs*(*id1*,*d*,*false*). The conditional statement in Robot1 indicates that after performing *rm* the same behavior is repeated with new position and direction parameters calculated by *NextPos* (see Eqn. (2)) and *NextDir* functions, respectively. On the other hand, the presence of obstacles only causes an in-place change of direction. The following expressions partially specify *NextDir*:

```
map NextDir:ID # Dir # Pnt # Bool -> Dir;
var cp:Pnt, cd:Dir, b:Bool;
eqn (cp!=PD1) -> NextDir(id1,cd,cp,false) =
  D(sgn(X(PD1)-X(cp)),sgn(Y(PD1)-Y(cp)));
  (cp!=PD1) -> NextDir(id1,cd,cp,true) =
  D(-Y(cd), X(cd));
  (cp==PD1) -> NextDir(id1,cd,cp,b) = D(0,0);
```

In this specification *sgn* is a function that extracts the sign of its argument and *PD1* is Robot1’s destination (Eqn. (4)). The notations *==* and *!=* denote data equality and inequality. Two instances *p1*,*p2* of *Pnt* are equal if and only if *X*(*p1*)=*X*(*p2*) and *Y*(*p1*)=*Y*(*p2*). The first and second rule of the equation system determine the next movement direction for the first robot when it is not at its destination. Absence of obstacles in the last scan activates the first rule and the next direction is calculated in order to guide the robot closer to its destination. Presence of an obstacle in the last scan activates the second rule which mimics a 90-degree counterclockwise rotation. The last rule sets the direction to (0,0) when the robot arrives at its destination. The complete specification can be derived by describing a similar behavior for *id2*.

Fig. 6 depicts the LTS described by Robot1 where each state is labeled by the data parameters carried by the process in that state.

**Environment** The Environment process records the position of the robots and performs the actions *es* and *em* in order to establish the required communications with the robot processes. The following mCRL2 syntax describes this process:

```
proc Environment(p1:Pnt,p2:Pnt) =
sum id:ID, d:Dir.es(id,d,Sense(id,p1,p2,d)).
  Sense(id,p1,p2,d)->
    Environment(p1,p2)<>
    (id==id1) ->
      em(id1,d).Environment(NextPos(p1,d),p2) +
    (id==id2) ->
      em(id2,d).Environment(p1,NextPos(p2,d));
```

The summations over the data types *ID* and *Dir* indicate that this process can establish a communication with any robot on the action *es* to perform a check for obstacles along direction *d*. The function *Sense* performs this check. If a movement is possible, Environment will update its information with the new position. Otherwise it will repeat the same

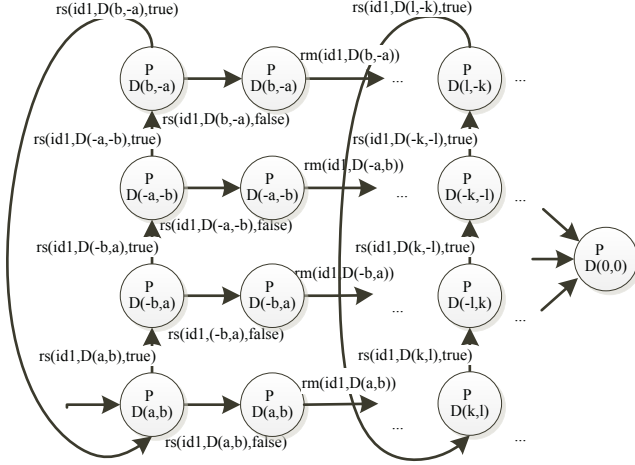


Figure 6: LTS description for Robot 1

behavior. A partial specification of Sense is as follows:

```

map Sense:ID # Pnt # Pnt # Dir -> Bool;
var cp1, cp2:Pnt, cd:Dir;
eqn (PD1!=cp1) -> Sense(id1, cp1, cp2, cd) =
    (cp2==NextPos(cp1, cd));
    (PD1==cp1) -> Sense(id1, cp1, cp2, cd) = false;

```

The first rule checks for presence of obstacles when the first robot is not in its destination. It simply checks whether a movement in the specified direction will cause a collision with the second robot (see Fig. 2). Since robots can only move along (0, 0) after arriving at their destinations, the second rule always declares absence of obstacles when Robot 1 arrives at its destination. The complete specification can be derived by describing a similar behavior for id2.

Finally, we initialize the specified processes and put them in parallel to achieve a model for the system (Fig. 5). To this end we use the following mCRL2 syntax:

```

init
allow({m, s}, comm({rm|em -> m, rs|es -> s},
    Robot1(P01, D01) || Robot2(P02, D02) ||
    Environment(P01, P02)));

```

The **comm** operator is used to establish communication between **rs** and **es** and renames this communication to a single action (**s**). The **allow** operator enforces the specified communications. In other words it blocks non-synchronous execution of **rs** and **es**. The actions **rm** and **em** are treated in the same way. The initial position and direction parameters, e.g., P01, can be specified similar to Eqn. (4).

## Verification

In what follows we first formalize the properties from *Validation Properties* in the modal  $\mu$ -calculus. We report on the results and observations we achieved on verifying these

properties against the specification discussed in the previous section. We applied the mCRL2 toolset for verification.

**Deadlock-freeness** In any reachable state of the system it is possible to perform an action:

$$[true^*](true)true \quad (7)$$

**Collision-freeness** Trajectories calculated by the algorithm will not cause a collision for two robots initially at P01 and P02, i.e., robots will not share a cell on the grid:

$$\begin{aligned}
 &\nu X(p_1 : Pnt = P01, p_2 : Pnt = P02). \\
 &((\forall id : ID, d : Dir. \neg m(id, d)]X(p_1, p_2)) \wedge \\
 &(\forall d : Dir. [m(id1, d)]X(NextPos(p_1, d), p_2)) \wedge \\
 &(\forall d : Dir. [m(id2, d)]X(p_1, NextPos(p_2, d))) \wedge \\
 &(p_1 \neq p_2)
 \end{aligned} \quad (8)$$

**Reachability** Robot1 (initially at P01) should reach its destination (PD1) with a finite number of movements:

$$\begin{aligned}
 &\mu X(p : Pnt = P01). \nu Y. ((\nexists d : Dir. m(id1, d)]Y \wedge \\
 &(\forall d : Dir. [m(id1, d)]X(NextPos(p, d)))) \vee (p == PD1))
 \end{aligned} \quad (9)$$

Applying the mCRL2 toolset we verified these properties against the model of the system. The properties (7) and (8) hold for any combination of different initial and destination positions for the robots chosen from the following set:

$$TestPoints = \{(x, y) | x, y \in \{0, \dots, 5\}\} \quad (10)$$

However, reachability does not hold in general. For instance, we identified the counterexample in Fig. 7. Movements of the first and second robot are depicted by filled and dashed arrows, respectively. The numbers denote the order of the performed moves. Initial and destination cells are marked by circles and flags, respectively. In this case the second robot moves relatively slowly compared to the first robot and it stops at the destination of the first robot. This causes a livelock in the first robot's behavior, i.e., it will perform the same sequence of actions without making any progress.

In an attempt to characterize the occurring problem we introduce a new notion of reachability where infinite movement steps are allowed when at least one of the robots is "close" to the destination of the other robot. The following formula formalizes this property:

$$\begin{aligned}
 &\mu X(p_1 : Pnt = P01, p_2 : Pnt = P02). \\
 &\nu Y(p'_1 : Pnt = p_1, p'_2 : Pnt = p_2). \\
 &(((\forall d : Dir. [m(id1, d)] \\
 &(((Near(p'_2, PD1) \vee Near(NextPos(p'_1, d), PD2)) \wedge \\
 &Y(NextPos(p'_1, d), p'_2)) \\
 &\vee (!Near(p'_2, PD1) \wedge !Near(NextPos(p'_1, d), PD2))) \wedge \\
 &X(NextPos(p'_1, d), p'_2))) \wedge \\
 &(\forall d : Dir. [m(id2, d)]Y(p'_1, NextPos(p'_2, d))) \wedge \\
 &([\nexists id : ID, d : Dir. !m(id, d)]Y(p'_1, p'_2))) \vee \\
 &(p_1 == PD1))
 \end{aligned}$$



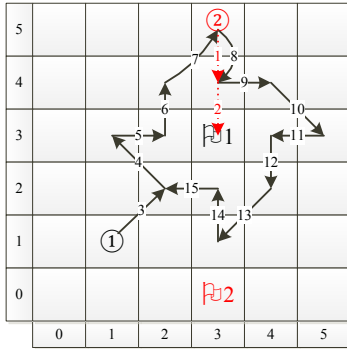


Figure 7: A counterexample for reachability

The function Near is defined as follows where abs is the built-in absolute value function:

```
map Near:Pnt # Pnt -> Bool;
var p1,p2:Pnt;
eqn Near(p1,p2) = (abs(X(p1)-X(p2)) <= 2 &&
                  abs(Y(p1)-Y(p2)) <= 2 );
```

This property is satisfied by the system described in the previous section for any reasonable combination of initial and destination positions from Eqn. (10).

## Conclusion

In the context of high level algorithmic approach to the path planning problem we studied ways to analyze multi-robot systems in a systematic way. The final goal is to verify a set of desired properties against a high-level model of a system and provide efficient feedback to the designer of the system.

We have introduced an approach based on process algebras for modeling and analyzing path planning algorithms. We have used the mCRL2 language and the modal  $\mu$ -calculus for describing a multi-robot system and its properties. The mCRL2 toolset has been used for the verification of these properties. We have applied this approach to investigate the correctness of a set of useful properties in a simple multi-robot system. Our observations show that for a simple path planning algorithm useful properties can be verified with the proposed approach efficiently (about one minute for each property on a standard desktop computer).

We envisage applying our approach to systems with more sophisticated path planning algorithms. Applying this approach to systems consisting of a large number of robots can also be considered as future work.

## References

- Baier, C. and Katoen, J.-P. (2008). *Principles of Model Checking*. MIT Press.
- Bristeau, P. J., Callou, P. J., Vissiere, D., and Petit, N. (2011). The navigation and control technology inside the AR.Drone micro UAV. In *Proceedings of IFAC'11*, pages 1477–1484.

- Cranen, S., Groote, J. F., Keiren, J. J., Stappers, F. P., de Vink, E. P., Wesselink, W., and Willemse, T. A. (2013). An overview of the mCRL2 toolset and its recent advances. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 199–213. Springer.
- Fainekos, G. (2011). Revising temporal logic specifications for motion planning. In *Proceedings of ICRA'11*, pages 40–45.
- Fainekos, G., Kress-Gazit, H., and Pappas, G. (2005). Temporal logic motion planning for mobile robots. In *Proceedings of ICRA'05*, pages 2020–2025.
- Groote, J. and Mateescu, R. (1999). Verification of temporal properties of processes in a setting with data. *Algebraic Methodology and Software Technology*, pages 74–90.
- Groote, J., Mathijssen, A., Reniers, M. A., Usenko, Y., and van Weerdenburg, M. (2009). Analysis of distributed systems with mCRL2. *Process Algebra for Parallel and Distributed Processing*.
- Jain, S., Sawlani, M., and Chandwani, V. (2010). Ad-hoc swarm robotics optimization in grid based navigation. In *Proceedings of ICARCV'10*, pages 1553–1558.
- Kress-Gazit, H., Fainekos, G., and Pappas, G. (2009). Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, 25(6):1370–1381.
- Laumond, J. P. (1998). *Robot Motion Planning and Control*. Springer-Verlag.
- Luigi Biagiotti, C. M. (2009). *Trajectory Planning for Automatic Machines and Robots*. Springer.
- min Han, K. (2007). Collision free path planning algorithm for robot navigation problem. Master's thesis, University of Missouri-Columbia.
- Mondada, F., Guignard, A., Bonani, M., Bar, D., Lauria, M., and Floreano, D. (2003). SWARM-BOT: from concept to implementation. In *Proceedings of IROS'03*, volume 2, pages 1626–1631.
- Pamosoaji, A. and Hong, K.-S. (2011). Collision-free path and trajectory planning algorithm for multiple-vehicle systems. In *Proceedings of RAM'11*, pages 67–72.
- Quottrup, M., Bak, T., and Zamanabadi, R. (2004). Multi-robot planning : a timed automata approach. In *Proceedings of ICRA'04*, volume 5, pages 4417–4422.
- Saska, M., Macas, M., Preucil, L., and Lhotska, L. (2006). Robot path planning using particle swarm optimization of Ferguson splines. In *Proceedings of ETFA'06*, pages 833–839.



# Autocatalysis Before Enzymes: The Emergence of Prebiotic Chain Reactions

Nathaniel Virgo, Takashi Ikegami

Ikegami Laboratory, University of Tokyo  
nathanielvirgo@gmail.com

## Abstract

How could complex, enzyme- or ribozyme-like molecules first have arisen on planet Earth? Several authors have suggested *autocatalytic cycles* as a partial answer to this question, since such reactions exhibit the life-like property of exponential growth while being composed of relatively simple molecules. However, a question remains as to the likelihood of an autocatalytic cycle forming spontaneously in the absence of highly specific catalysts. Here we show that such cycles form readily in a very simple model that includes no direct catalysis reactions. Catalytic effects nevertheless emerge as properties of the reaction network. This suggests that the conditions for the formation of such cycles are not difficult to achieve. The resulting cycles solve the problem of specificity not by being small and simple but by being large and complicated, suggesting that early prebiotic metabolisms could have been extremely complex. We predict that this phenomenon can be reproduced in wet chemistry. We discuss the challenges involved in this, as well as the implications for how we view the origins of life.

## Introduction

A necessary requirement for biological metabolism is *autocatalytic kinetics*, i.e. the ability of a set of chemical species to increase its own rate of production. Without the ability to positively influence the production of its own chemical components, the prebiotic equivalent of a living organism would be able neither to reproduce nor to maintain its own composition over time. In this paper we investigate the possibility that the earliest proto-metabolisms achieved this through a mechanism known as an autocatalytic cycle.

In this paper we present a highly simplified model of a simple organic polymer chemistry operating away from thermodynamic equilibrium. This model is extremely simple, consisting only of basic synthesis and decomposition reactions, with no catalytic kinetics assumed *a priori*. We find that autocatalytic cycles form readily in such a system, suggesting that the chemistry in which the first steps toward metabolism took place could have been much simpler than generally supposed. The networks that emerge in our model are complex, consisting of many interlinked catalytic and autocatalytic cycles. The highly interconnected nature of

these autocatalytic subnetworks means that a reaction involving one of the intermediates is likely to produce another intermediate, thus overcoming the much-discussed problem of specificity in autocatalytic cycles. This suggests that complex autocatalytic reaction networks formed from simple molecules can be produced much more easily than simple networks composed of complex “replicator” molecules.

Because the requirements for this phenomenon are so easy to meet, it should be possible to observe it experimentally, in prebiotic chemistry experiments along the lines of the Miller-Urey experiment or HCN polymerisation. To achieve this one would need to change the conditions so that the breakdown of polymers via hydrolysis or oxidation occurs in the same system as their synthesis, at a comparable rate. This simultaneous build-up and break-down of polymers is analogous to anabolism and catabolism in biology. We comment on the potential implications of such a result, and the challenges that would be involved in attaining it.

It is worth pointing out a major difference between our model and one of the predominant existing approaches to explaining the origin of biological autocatalysis. As discussed below, there are many studies that model the emergence of autocatalysis in networks of reactions between peptide or RNA-like molecules, via a mechanism known variously as reflexive autocatalysis, autocatalytic sets or RAF sets. This work has shown that autocatalysis is easy to achieve via this mechanism even if the reaction networks are chosen at random rather than having autocatalysis “designed” into them (Kauffman, 1986); and that such autocatalytic sets are capable of evolution by natural selection via an attractor-based heredity mechanism, even in the absence of specific information-carrying molecules (Vasas et al., 2012).

However, this definition of autocatalysis presupposes the existence of single-step catalysis reactions, and therefore entails an assumption that the molecules involved are complex enough to behave as enzymes. Because our aim is to explain the emergence of such complex molecules from simpler reactants, we focus instead on a different mechanism: the *autocatalytic cycle* or *branching chain reaction* (King, 1978).

For our purposes, a branching chain reaction may be de-

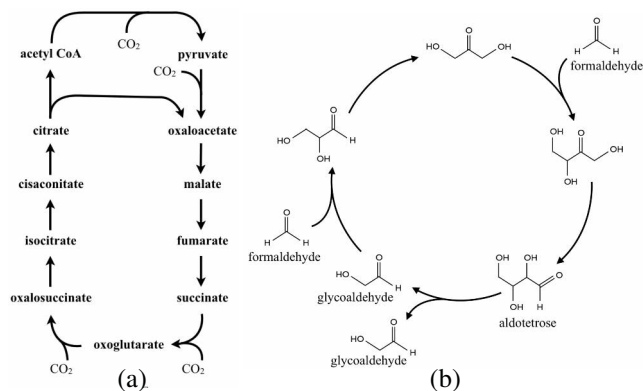


Figure 1: (a) Schematic of the reductive citric acid cycle, redrawn from Morowitz et al. (2000). The branching step is the splitting up of citrate into oxaloacetate and acetyl CoA, which is then transformed into a second oxaloacetate, so that its concentration doubles on every turn of the cycle. (b) The mechanism of the formose reaction, as proposed by Breslow (1959). The branching step is the decomposition of an aldohexose into two molecules of glycolaldehyde. The formose reaction has been observed experimentally, without the use of biological catalysts.

defined as a net chemical reaction, at least one of whose products is also an intermediate. This allows the concentration of intermediates to build up over time, which under the right conditions can lead to exponential growth. Such reactions are not uncommon and are often the mechanism behind combustion and explosive reactions. A more formal definition of this type of autocatalysis is given by Andersen et al. (2012). In the classification of Plesson et al. (2011), this definition includes direct, indirect and autoinductive forms of autocatalysis.

Some known examples of autocatalysis via branching chain reactions are shown in Fig. 1 and 2. This definition is similar in spirit to that of an autocatalytic set, but in our case the catalysis mechanism emerges from the system's dynamics, rather than being a property of individual molecules.

Autocatalytic cycles have been hypothesised as playing an important role in the origins of life. Wächtershäuser (1988), and later Morowitz et al. (2000) proposed the *reductive citric acid cycle* (Figure 1a) as a possible means by which molecules such as sugars, lipids and amino acids could have been generated on the early Earth. The citric acid cycle is important in modern biology but its intermediate steps are catalysed by enzymes. Wächtershäuser's argument was that inorganic surface catalysts might have been able to play the same role on the early Earth. Morowitz et al. argued that the reductive citric acid cycle might be unique, in the sense of being the only autocatalytic cycle that could lead to the complexity of modern life on an Earth-like planet.

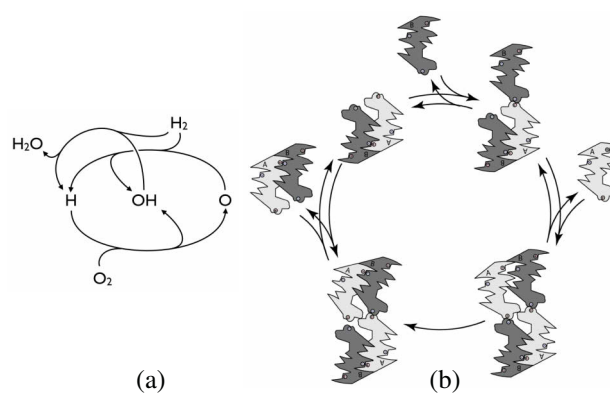


Figure 2: Some other known examples of autocatalysis via chain reactions. (a) A few of the most important reaction steps in the early stages of the combustion of H<sub>2</sub>, demonstrating autocatalysis via a more complex network than a single cycle. H<sub>2</sub> and O<sub>2</sub> can be mixed without reacting, but due to this mechanism they will react very rapidly after an initial spark produces small quantities of H and O. (b) Template replication is a special case of chain reaction autocatalysis. Here, an AB dimer catalyses the formation of another AB dimer through complementary base pairing. Figure 2b is taken from Virgo et al. (2012), in which a physical instantiation of template replication was demonstrated using macroscopic “monomers” floating above an air table.

These ideas have been criticised on the grounds that it would be difficult to find mineral catalysts that would catalyse every step in this relatively complex cycle (Orgel, 2000) without also catalysing side-reactions that would reduce the replicator's specificity to a non-viable level (Szathmáry, 2000). This latter problem must be solved by any approach to the origins of life. In any autocatalytic chemical system there will be reactions that contribute to the autocatalytic network (branching reactions and propagating reactions) and reactions that deplete its constituents (terminating reactions). If the latter dominate then growth will not occur.

In this paper we offer solutions to these problems. King (1982) gave a heuristic argument that the formation of autocatalytic cycles is very likely in systems that are driven by a flow of energy across their boundary but closed to matter flow. This is because the products of any reaction will eventually be recycled, and this recycling process has a high probability of forming part of an autocatalytic cycle. Our model confirms that this phenomenon can occur in very simple driven systems, even if the system is not completely closed to matter flow. This suggests that there may be a great number of simpler autocatalytic systems that could have preceded the reductive citric acid cycle, perhaps ultimately leading to the production of complex organic molecules that could play the role of enzymes.

In the same paper, King argued that autocatalytic cycles with many intermediate species are statistically unlikely to be viable, in the sense of being able to grow exponentially. This is because every step in an autocatalytic cycle is vulnerable to side reactions. Every reaction step may be assigned a number between 0 and 1 representing its specificity, and it can be shown that the cycle is only viable if the product of the specificities passes a threshold. Hence, all else being equal, a cycle with many steps is less likely to be viable than one with only a few. However, in our model we observe fairly large autocatalytic systems that are composed not of a single cycle but of many intersecting catalytic and autocatalytic cycles. It seems that such systems avoid the need for specificity simply by including such a large number of species that the production of molecule that *isn't* part of the autocatalytic network is comparatively low.

Our model shows that branching chain reactions occur rather easily under certain conditions. Essentially all that is needed is the simultaneous presence of synthesis reactions (such as polymerisation) and decomposition reactions (such as oxidation or hydrolysis), as well as a source of free energy that causes some reactions to be favoured over others. These processes are closely analogous to anabolism and catabolism in a living cell. On the early Earth there were a wide variety of potential energy sources (Deamer and Weber, 2010) as well as, presumably, a wide variety of environments of varying temperatures, pressures, pH values, redox conditions etc., making it fairly likely for such conditions to be satisfied somewhere on the planet. Such conditions should also be relatively easy to achieve experimentally.

Below we survey the two main existing approaches to the emergence of autocatalysis within the field of ALife, before presenting our own model and its results. This is followed by an extended discussion of how this phenomenon fits into our picture of the early Earth, as well as the challenges involved in demonstrating it in wet chemistry experiments.

### Artificial Chemistry approaches to Autocatalysis

Our aim in this work is to apply an “artificial chemistry” methodology to the question of how autocatalytic cycles can arise in prebiotic chemistry. In this section we briefly survey previous work that has had similar aims. This previous work has two main starting points: the work of Kauffman (1986) and the work of Fontana and Buss (1994).

A central work in the metabolism-first school of the origins of life is the model of Kauffman (1986), who showed that, even if the reaction scheme of an artificial chemistry is chosen completely at random, the probability of a collectively autocatalytic set of protein-like polymers becomes high as the number of species present increases. This is an important idea, because it implies that under the right circumstances, the emergence of something akin to biological metabolism might be almost inevitable, even without the organising force of natural selection. With good reason, this

work has spawned a multitude of successors.

However, it must be stressed that, due to its origins in a theory of protein interactions, this body of work assumes a particular mechanism for autocatalysis, which can only occur in relatively complex chemistries. This mechanism relies on the idea that the molecules involved are each able to behave like enzymes, selectively catalysing some reactions but not others in a way that can be modelled as a single-step reaction. This would require the monomers to be of a certain level of complexity. Our aim is to show that similar phenomena can occur without assuming enzyme-like kinetics.

From a quite different direction, the work of Fontana and Buss (1994) looked for autocatalysis in chemistries where the molecules were represented as Lambda calculus expressions. The goal of this work was to investigate the generation of novelty through the formation of autocatalytic structures. This work also spawned a large number of successors, including the work of Ikegami and Hashimoto (1995), who looked for the emergence of autocatalysis in networks of Turing machines and tapes under a noisy environment.

Work in this sub-field tends not to include thermodynamic considerations, choosing instead to emphasise the structure of the reaction network itself. A secondary goal of our work is to investigate the impact of thermodynamic considerations on such “abstract chemistries”. In particular, in real chemistry, reactions may proceed in the forward or in the reverse direction, depending on the free energy difference between the reactants and the products. We will argue that giving the system the ability to “choose” the direction of reactions in this way is important for the emergence of autocatalysis.

### A Simple Model

We are concerned with the question of whether autocatalytic cycles, or more complex branching chain reactions, can occur in simple (non-enzymatic) organic chemistry. To do so we use a model in which a reaction network is randomly generated by allowing or disallowing cleavage and ligation reactions between polymers. A key difference between our work and previous work is that in our model no polymer can directly catalyse any reaction, so any autocatalysis that occurs must be via cycles rather than enzyme-like catalysis.

In polymer models in artificial chemistry, molecules are usually considered to consist of a string of  $m$  different types of monomer. For the sake of simplicity, in this work we set  $m = 1$ , restricting ourselves to a single monomer type, denoted A. The possible species can therefore be written  $A_1, A_2, \dots, A_n$ , where  $n$  is a maximum allowed polymer length, imposed for reasons of computational tractability. These are intended to represent molecules based on simple carbon chains, rather than complex heteropolymers such as peptides or RNA strands.

All reactions must preserve the number of monomers. We consider only reactions of the form  $A_i + A_j \rightleftharpoons A_k$ , where  $k = i + j$  is not greater than  $n$  and, to avoid duplicates,

$i \leq j$ . To generate a reaction network we must decide, for every such reaction, whether to include it in the network or not. For simplicity, following Kauffman, we simply include each reaction in the network with a constant uniform probability  $p$ , independently of every other reaction. The double arrow indicates that the forward reaction  $A_{i+j} \longrightarrow A_i + A_j$  and the reverse reaction  $A_i + A_j \longrightarrow A_{i+j}$  are always either both included in the network or both not included. This is required for consistency with thermodynamics, and as we will see, it plays an important role in the emergence of autocatalytic networks in our model.

We assume that the rate constants of all the included reactions are equal. Setting the forward rate constant to 1 without loss of generality and letting  $k$  stand for the reverse rate constant, we let each reaction  $A_i + A_j \rightleftharpoons A_{i+j}$  occur at a net rate  $R_{ij} = a_i a_j - k a_{i+j}$ , where  $a_i$  is the molar concentration of species  $i$ .  $R_{ij}$  may be positive or negative, representing a net synthesis or net decomposition reaction, depending on the concentrations of the three reactants. The justification for the  $a_i a_j$  term is that the ends of two polymer molecules must meet in order for them to undergo a ligation reaction, and we assume that the polymer tips, rather than the polymers themselves, behave like point particles in a well mixed system. (Such “mass action” assumptions are common in models of polymerisation kinetics.) The  $-k a_{i+j}$  term simply means that, for every decomposition reaction in the network, there is a constant probability per unit time that it will occur in a given molecule. We write  $R_{ij} = 0$  if the reaction is not included in the network.

This leads to the following set of dynamical equations:

$$\dot{a}_i = \phi_i + \sum_{k=1}^{i-1} R_{k,i-k} - \sum_{j=1}^i R_{ij} \quad (1)$$

where the  $R_{ij}$  are as defined above, and  $\phi_i$  represents the flux of  $A_i$  in or out of the system, as explained below. The two summation terms arise from the fact that each species  $A_i$  is involved in reactions of the form  $A_k + A_{i-k} \rightleftharpoons A_i$  as well as  $A_j + A_i \rightleftharpoons A_{i+j}$ .

### Thermodynamic properties

If we let the fluxes  $\phi_i = 0$ , the system will approach thermodynamic equilibrium. In such a state the reactions have the property of *detailed balance*, meaning that the forward and reverse rates are equal for every reaction. For a reaction  $A_i + A_j \rightleftharpoons A_{i+j}$  this occurs when  $a_i a_j = k a_{i+j}$ , or  $\log a_i + \log a_j = \log k + \log a_{i+j}$ . We may therefore define the *chemical potential*  $\mu_i$  of species  $i$  to be  $(i-1) \log k + \log a_i$ . This has the property that when the system is in thermodynamic equilibrium,  $\mu_i + \mu_j = \mu_{i+j}$ . (The usual thermodynamic definition of chemical potential would include a factor of  $RT$ , the gas constant times the temperature, which we have set to 1 for convenience.) From this we may define the Gibbs energy  $G = \sum_i \mu_i a_i$ . In accordance

with the second law,  $G$  cannot increase over time unless we allow some fluxes of matter in and out of the system. For a closed system,  $G$  is a Lyapunov function.

If we temporarily assume that every allowable reaction is included in the network ( $p = 1$ ), we can see that the equilibrium concentrations must satisfy  $a_i = k e^{C_i}$ , for some constant  $C$ , in order for detailed balance to hold for every reaction. The value of the constant  $C$  depends on the initial conditions, which stems from the fact that the total number of monomers in the system,  $\sum_i i a_i$ , is conserved. Low initial concentrations will lead to decomposition reactions being favoured, and therefore low (negative) values for  $C$ , whereas high total monomer concentrations lead to synthesis reactions being favoured. High enough concentrations lead to positive values for  $C$ , meaning that the equilibrium conditions are dominated by the longest possible polymers rather than by short ones. This phenomenon is observed in real polymer chemistries. If  $p < 1$  then it is possible for equilibrium situations to exist where this condition is not satisfied, because conservation laws arise that prevent some concentrations from becoming equilibrated with one another. However, in these cases higher concentrations still lead to longer products being favoured.

In order to observe the operation of autocatalytic cycles, the system must be held away from thermodynamic equilibrium. In real chemical systems this can be achieved in many ways. For example, by cycling the temperature or pH (both of which would effectively change  $k$  in our model), or through electrochemistry or photochemistry, which can drive reactions that would otherwise not be thermodynamically favourable. In the first set of results below we model the reactions as being held out of equilibrium by continually adding reactants and removing products, as in a flow reactor; in the second we simply start the system in an initial state far from equilibrium and observe the decrease in Gibbs energy over time.

## Results

In this section we present the results from two different simulations based on the above model. The first serves as a useful demonstration of the formation of autocatalytic cycles in driven systems, but is somewhat contrived; the second shows that autocatalytic kinetics can arise in larger, randomly-generated systems.

In our first model we set  $p = 1$ , including every reaction in the network, but we limit the size of the largest polymer. There are many ways in which the system may be held out of equilibrium; in this illustrative example we do it by letting the fluxes  $\phi_1$  and  $\phi_2$  have nonzero values, with their rates chosen such that the concentrations  $a_0$  and  $a_1$  are held constant at 100 and 0.1 respectively. Conceptually,  $A_1$  flows into the system, then undergoes a series of reactions until it is converted into  $A_2$ , at which point it is removed. Boundary conditions of this type could be achieved experimentally



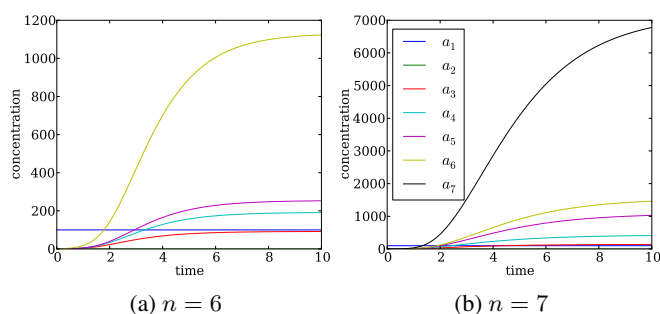


Figure 3: Time series of the concentrations in the model when  $n = 6$  and  $n = 7$ . In both cases there is a period of exponential growth from  $t = 0$  to about  $t = 2.0$ , indicating autocatalysis. This is followed by gradual saturation.

using membrane permeable only to small molecules.

Figure 3 shows the dynamics of this system when  $n = 6$  and  $n = 7$ . In both cases there is a period of exponential growth followed by a period of saturation. Exponential growth is a key experimental sign of autocatalysis. (With  $n \leq 5$  this effect does not occur.)

Figure 4 shows the reaction networks that arise once these systems have reached a steady state. (We were unable to find more than one attractor in these particular systems, although the existence of others cannot be ruled out.) The recycling structure of these networks can be seen as a response to the flux of Gibbs energy across the system's boundary, in accordance with Morowitz' (1966) cycling theorem. It is for this reason that we believe including thermodynamically realistic kinetics in such models is important for understanding the origins of autocatalytic cycles.

In both cases the mechanism behind the exponential growth is an autocatalytic cycle that produces two molecules of an intermediate for every molecule present initially; this exponential growth is countered by decay reactions once the concentrations become high. However, the two systems use different autocatalytic cycles. This is possible because the direction in which reactions occur is determined by the differences in the reactants' chemical potentials, and these depend upon the system's dynamics.

As  $n$  is increased further, more catalytic and autocatalytic cycles emerge (results not shown). However, it can be seen from Figure 3 that the concentrations of longer polymers are much higher than those of short ones; this trend continues as  $n$  is increased, leading to unrealistic results as  $n$  becomes large, since in reality a system composed mostly of long polymers will become viscous or solid, preventing further reactions by suppressing mixing.

However, this issue can be resolved by choosing different values for the parameters, so that shorter rather than longer ones are thermodynamically favoured. In addition to doing this we set  $n$  large enough that the longest polymer only

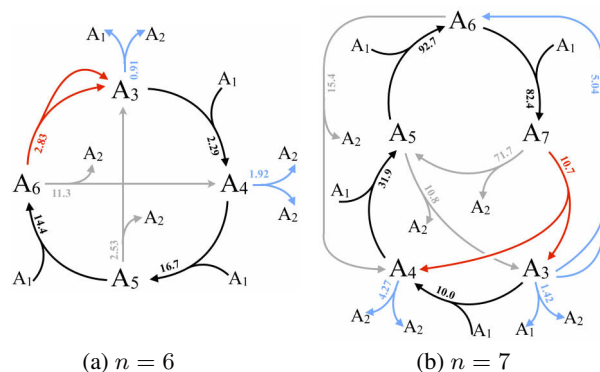


Figure 4: The reaction networks that form when  $n = 6$  and  $n = 7$ , with  $p = 1$  and the concentrations of  $A_1$  and  $A_2$  held constant. Propagating reactions are shown in black or grey, branching reactions in red, and terminating reactions in blue. Numbers represent the rate at which each reaction occurs once a steady state is reached, in multiples of  $10^{-3}$  concentration units per time unit. The set of allowed reactions is predetermined, but the direction in which they proceed depends on the system's dynamics. Both networks contain several catalytic cycles, coupled to an autocatalytic cycle (highlighted in black). The reaction  $A_6 \rightarrow 2A_3$  is the key branching step when  $n = 6$ , but when  $n = 7$  it runs in the opposite direction, becoming a depleting reaction.

ever exists at a low concentration. The parameters we use are  $K = 100$ ,  $p = 0.2$  and  $n = 40$ . When such a system is driven toward a steady state, it produces very complex networks that are difficult to analyse. Andersen et al.'s (2012) algorithm could be used to detect autocatalytic subnetworks, but it cannot tell us how viable they are. Because of this, instead of driving the system we simply initialise it in a state with a high Gibbs energy and observe its return to equilibrium. This allows us to detect autocatalysis by observing exponential growth in the kinetics. We use the initial conditions  $a_1 = 1000.0$ , and  $a_i = K^{1-i}$  for  $i > 1$ . This can be interpreted as a system that was initially in equilibrium, to which a large quantity of monomers has just been added.

Figure 5 shows the results of this simulation. We numerically integrated the dynamics of 50 randomly generated networks for 3 time units each. In 32 out of the 50 cases, no reactions occurred and the system remained in its initial state. In 14 out of the remaining 18 cases, there was at least one period of time in which a species' concentration increased with  $d^2a_i/dt^2 > 0$  and  $da_i/dt > 0.01$ . Such "accelerating" growth is an indication that there is a viable autocatalytic network within the system.

The behaviour of the system is quite sensitive to the choices of parameters, but the phenomenon of exponential growth appears to be fairly robust. Quantifying this is a task for future work.



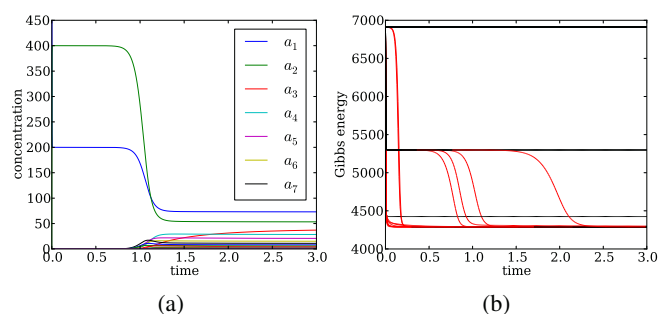


Figure 5: (a) An example of the dynamics when  $n = 40$ ,  $p = 0.2$  and the system is closed but initially out of equilibrium. In this case the reaction  $2A_1 \rightleftharpoons A_2$  rapidly goes to equilibrium and the system stays in this state for a while before a complex autocatalytic network arises and rapidly brings the system near to thermal equilibrium. (b) Superimposed results from 50 networks, showing the change in Gibbs energy over time. The initial conditions are identical for each network and have a Gibbs energy of about 6900, whereas the equilibrium state has a Gibbs energy of about 4300. The lines are coloured red when at least one species in the system is increasing with a positive second time derivative, indicating the presence of a viable autocatalytic network. The systems typically approach equilibrium more rapidly when an autocatalytic network is operating.

## Discussion and Future Work

We have presented a model that couples a simple abstract chemistry with thermodynamically realistic kinetics, in order to show that autocatalysis via branching chain reactions can occur even in very simple chemical systems. The origins of life are often thought of in terms of a “self-replicating molecule” which, as Figure 2b shows, can be thought of as a small autocatalytic cycle composed of reactions between complex molecules. Our results suggest that it may be much easier to achieve the opposite: a large autocatalytic network composed of simple molecules.

An important property of our model is that both forward and reverse reactions are included, subject to thermodynamically realistic kinetics. As a result of this we can observe that an externally introduced source of energy drives cycling behaviour (Morowitz, 1966), and this recycling leads to autocatalytic kinetics (King, 1982). We therefore believe that adding reverse reactions and thermodynamic constraints to “abstract chemistry” models along the lines of (Fontana and Buss, 1994) could shed light on the process of novelty generation in general, as well as the origins of life in particular.

The main components of our model are (i) a system that is at least partially closed to matter flow, in which both synthesis and decomposition reactions can occur; and (ii) an energetic driving force, which causes some reactions to be

favoured over others. The simplicity of our model suggests that such conditions are essentially all that is required for autocatalytic networks to form. This makes it much more plausible that autocatalytic chemical systems could emerge on the early Earth, and the simplicity of the conditions makes the idea amenable to experimental testing in real chemistry.

It has been shown that Kauffman’s autocatalytic sets are capable of evolution by natural selection, even without the existence of specific information-carrying molecules (Vasas et al., 2012). Our hope is that something similar will be true of autocatalytic systems that occur via chain reactions rather than single-step enzyme-like catalysis. If this is the case then we may suggest that life did not start with the citric acid cycle but with a different autocatalytic system, perhaps composed of simpler molecules, but forming a much more intricate network of reactions. The catalysts required to produce the molecules of modern life via the reductive citric acid cycle could then have been arrived at by natural selection acting on the original autocatalytic system.

However, the models we have presented here seem not to exhibit the large number of attractors that would enable heredity in such a way. We must therefore discuss what additional conditions might need to be met in order for an evolvable system to arise.

## Constraints on the Reaction Network

Our model obeys constraints imposed by mass conservation and the laws of thermodynamics, but beyond this we choose the permitted reactions at random. As we have seen, this results in autocatalytic networks that tend to include almost every possible species as part of their network. In order for the system to exhibit a large number attractors there would need to be multiple possible autocatalytic networks capable of out-competing each other.

Real chemical reaction networks are not random but are determined by the physics of molecular interactions. This imposes a number of constraints both on the topology of chemical reaction networks and on their kinetics, and such constraints might help to “partition” the network into multiple potential autocatalytic sub-networks. Perhaps the most obvious such constraint is imposed by stoichiometry: chemical reactions must conserve not only mass but also the number of nuclei of each chemical element, as well as electrons. Our system recycles monomers, but in biology (particularly at the ecosystem level) the recycling of specific nutrients such as nitrogen and phosphorous plays an important organising role. We therefore suspect that adding multiple conservation laws to our model will enable a richer range of behaviours than it currently exhibits. (However, this would greatly increase the number of possible molecular species in the model, requiring a change in modelling methodology from the simple ode integration that we have used here.)

Another important set of constraints are given by the shapes of molecules and the ways in which they interact

electromagnetically. This gives reaction networks the important property that similar molecules can undergo similar reactions. Modelling the relationship between the form of molecules and the resulting reaction network is of course rather difficult, but perhaps something like the approach of Fontana and Buss (1994), combined with the thermodynamic realism of the present model, would be a useful tool to investigate this question.

Finally, the existence of phase changes can also put constraints on the reaction network. King (1982) argued that this could enhance the formation of viable autocatalytic cycles. Adding phase separation to our model would allow us to investigate this idea.

### **The Importance of Spatial Structure and Compartmentalisation**

In addition to phase separation, more complex spatial structuring may be important in going from simple to more complex autocatalytic networks. Many previous studies, including some by the present authors, have concluded that spatial self-organisation is important for avoiding “parasitic” side-reactions, i.e. sets of reactants that produce themselves autocatalytically, feeding not directly on the energy source but on the original autocatalyst (e.g. Boerlijst and Hogeweg, 1991; Froese et al., 2011, 2012, 2013). In (Froese et al., 2012; Virgo et al., 2013) we found that in a spatial context, parasitic reactions could become a positive benefit to the primary autocatalytic system, leading to evolvability. We expect that embedding a system along the lines of the present model in a spatial context will lead to richer dynamics.

Many hypotheses about the origins of life require “compartmentalisation”, the formation of a lipid vesicle, or similar small compartment, in which reactions take place. One reason for this arises from energetics: for complex biological polymers such as peptides to form, the monomers must be present in sufficient concentration, and since large monomers like amino acids or RNA bases are difficult to produce in such concentrations, a membrane is required in order to prevent them from diffusing into the environment.

With simpler molecules the energetics of polymerisation are less constrained, and simple monomers could more easily be produced abiotically. Compartmentalisation is therefore less critical for the kind of prebiotic system we consider in this paper, and one can therefore imagine such phenomena occurring in a relatively dilute “prebiotic soup”, or more accurately, a prebiotic flow reactor.

A second reason to require compartmentalisation is simply that there must be a population of multiple individuals in order for natural selection to occur. We suggest that simple spatial separation could have played this role originally, in a manner outlined in (Virgo, 2011; Froese et al., 2012), only later to be replaced by membrane-bound cell structures.

If autocatalysis can occur in solution, and if the autocatalytic network also produces lipid-like molecules, then

membrane-bound protocells may be able to form spontaneously (Ono and Ikegami, 2000; Madina et al., 2003). This neatly solves the chicken-and-egg problem of how membrane-bound autocatalysis could first have arisen.

### **Towards Empirical Verification**

In our model, autocatalysis via branching chain reactions emerges in a system that contains only simple synthesis and decomposition reactions, together with a supply of free energy. This idea should be relatively easy to demonstrate experimentally. Previous experiments relevant to the origins of life, such as the Miller-Urey experiment Miller (1953) or the polymerisation of hydrogen cyanide (HCN) (see, e.g., Minard et al., 1998) have focused on the production of organic molecules through polymerisation. Both of these experiments produce a diverse mixture of products, including amino acids; however, these products form into a black, sticky “tar” called tholin that seems unlikely to self-organise into anything like a biological metabolism, despite the fact that tholin itself is thermodynamically unstable and can be used as an energy source by several common species of bacteria (Stoker et al., 1990).

Our results suggest that autocatalytic cycles may emerge in such experiments if the conditions are changed so that breakdown of polymers via hydrolysis or oxidation can occur simultaneously with the polymerisation, at a comparable rate. Since polymer molecules are continually built up and broken down, we would expect those that can produce themselves autocatalytically to persist at the expense of those that cannot. The kinetics and energetics of both polymerisation and depolymerisation are sensitive to environmental factors such as temperature, pH, monomer concentration and the presence of surfaces and inorganic catalysts. Achieving autocatalysis should simply be a case of setting the appropriate conditions for the reaction. We are currently working on demonstrating this in an HCN polymerisation experiment.

The challenge in such an experiment is in demonstrating that an autocatalytic cycle has indeed emerged. The sheer number of products means that the resulting mixture tends to have a continuous mass spectrum, making it difficult to identify which species are present. However, evidence for autocatalytic kinetics would be given by sudden changes in the mass spectrum, even if one cannot readily identify the species responsible.

### **The Prebiotic Ecosystem**

Above we have mentioned several phenomena, such as nutrient cycling and parasitism, that one would normally associate with physical ecology than purely chemical systems. It is worth drawing an explicit conclusion from this: we believe that prebiotic systems should be thought of as resembling ecosystems, complete with food chains, nutrient cycling, energetic restrictions and all the rest — everything except for clearly differentiated living cells, which arose later.

We know that the early Earth was a very active world, with sources of chemical free energy from UV photochemistry in the atmosphere, shockwaves from asteroid impacts, radioactivity, lightning, volcanoes and geochemistry (Deamer and Weber, 2010), and matter cycling due to plate tectonics and the water cycle. In such a context, it is easy to imagine that such prebiotic ecosystems could have been a global phenomenon, leading to primordial equivalents of today's biogeochemical cycling of nitrogen, phosphorous and carbon. From this point of view a homeostatically self-regulating Earth system should be seen not as a *consequence* of the biosphere (Lovelock, 1987), but rather as the context in which it first arose.

## Conclusion

We have set out to explain how autocatalysis could have emerged on the early Earth, before the existence of enzyme-like catalysts. We have shown, using a simple model, that autocatalytic cycles can emerge in chemical systems with only synthesis and decomposition reactions, without requiring the molecules to have special catalytic properties. The resulting autocatalytic networks solve the problem of specificity not by being small and simple but by being large and complicated. We conclude that the earliest origins of life may have lain not in a "minimal" autocatalytic system but in a "maximal" one.

The conditions required for this to occur are simple enough that we hope it can be demonstrated in wet chemistry experiments, and we have discussed how this could be achieved. Finally, based on our results, we have argued that the prebiotic Earth should be seen not as a soup but as an ecosystem.

## References

- Andersen, J. L., Flamm, C., Merkle, D., and Stadler, P. (2012). Maximizing output and recognizing autocatalysis in chemical reaction networks is np-complete. *Journal of Systems Chemistry*, 3(1):1–9.
- Boerlijst, M. C. and Hogeweg, P. (1991). Spiral wave structure in pre-biotic evolution: Hypercycles stable against parasites. *Physica D*, 48:17–28.
- Breslow, R. (1959). On the mechanism of the formose reaction. *Tetrahedron Letters*, 21:22–26.
- Deamer, D. and Weber, A. (2010). Bioenergetics and life's origins. *Cold Spring Harbour Perspectives in Biology*, 2(a004929).
- Fontana, W. and Buss, L. W. (1994). What would be conserved if "the tape were played twice"? *Proceedings of the National Academy of Sciences*, 91:757–761.
- Froese, T., Ikegami, T., and Virgo, N. (2012). The behavior-based hypercycle: From parasitic reaction to symbiotic behavior. In Adami, C. et al., editors, *Proceedings of Artificial Life 13*, pages 457–464. MIT Press.
- Froese, T., Virgo, N., and Ikegami, T. (2011). Life as a process of open-ended becoming: Analysis of a minimal model. In Lenaerts, T. et al., editors, *Advances in Artificial Life, ECAL 2011*, pages 250–257, Cambridge, MA. MIT Press.
- Froese, T., Virgo, N., and Ikegami, T. (2013). Motility at the origin of life: Its characterization and a model. *Artificial Life*, Early Access publication February 1st.
- Ikegami, T. and Hashimoto, T. (1995). Active mutation in self-reproducing networks of machines and tapes. *Artificial Life*, 2(3):305–318.
- Kauffman, S. (1986). Autocatalytic sets of proteins. *Journal of Theoretical Biology*, 119:1–24.
- King, G. A. M. (1978). Autocatalysis. *Chemical Society Reviews*, 7(2):297–316.
- King, G. A. M. (1982). Recycling, reproduction and life's origins. *BioSystems*, 15:89–97.
- Lovelock, J. (1987). *Gaia: a New Look at Life on Earth*. Oxford University Press.
- Madina, D., Ono, N., and Ikegami, T. (2003). Cellular evolution in a 3d lattice artificial chemistry. In Banzhaf, W. et al., editors, *Advances in Artificial Life: 7th European Conference, ECAL 2003*, pages 59–68.
- Miller, S. L. (1953). Production of amino acids under possible primitive earth conditions. *Science*, 117:528–529.
- Minard, R. D., Hatcher, P. G., and Gourley, R. C. (1998). Structural investigation of hydrogen cyanide polymers: New results using TMAH thermochemolysis/GC-MS. *Origins of Life*, 28:461–473.
- Morowitz, H. (1966). Physical background of cycles in biological systems. *Journal of Theoretical Biology*, 13:60–62.
- Morowitz, H., Kostelnik, J., Yang, J., and Cody, G. (2000). The origin of intermediary metabolism. *Proceedings of the National Academy of Sciences*, 97(14):7704–7708.
- Ono, N. and Ikegami, T. (2000). Self-maintenance and self-reproduction in an abstract cell model. *Journal of Theoretical Biology*, 206:243–253.
- Orgel, L. (2000). Self-organising biochemical cycles. *PNAS*, 97(23):12503–12507.
- Plesson, R., Brandenburg, A., Jullien, L., and Bersini, H. (2011). Autocatalysis: At the root of self-replication. *Artificial Life*, 17:219–236.
- Stoker, C. R., Boston, P. J., Mancinelli, R. L., Segal, W., Khare, B. N., and Sagan, C. (1990). Microbial metabolism of tholin. *Icarus*, 85(1):241–256.
- Szathmáry, E. (2000). The evolution of replicators. *Philosophical Transactions of the Royal Society, Series B*, 355:1669–1676.
- Vasas, V., Fernando, C., Santos, M., Kauffman, S., and Szathmáry, E. (2012). Evolution before genes. *Biology Direct*, 7(1).
- Virgo, N. (2011). *Thermodynamics and the Structure of Living Systems*. PhD thesis, University of Sussex, UK.
- Virgo, N., Fernando, C., Bigge, B., and Husbands, P. (2012). Evolvable physical self-replicators. *Artificial Life*, 18:129–142.
- Virgo, N., Froese, T., and Ikegami, T. (2013). The positive role of parasites in the origins of life. To appear in proceedings of IEEE SSCI 2013.
- Wächtershäuser, G. (1988). Before enzymes and templates: Theory of surface metabolism. *Microbiology and Molecular Biology Reviews*, 52(4):452–484.

# Synthetic signalling protocell networks as models of neural computation

Matthew D. Egbert<sup>1,2,\*</sup>, Gerd Grünert<sup>1</sup>, Gabi Escuela<sup>1</sup> and Peter Dittrich<sup>1</sup>

<sup>1</sup> Biosystems Analysis Group, Friedrich Schiller University, Jena, Germany

<sup>2</sup> Center for Computational Neuroscience and Robotics, University of Sussex, Brighton, U.K.

\* mde@matthewegbert.com

## Introduction

Modern conventional computers are programmable, predictable and relatively easy to understand and engineered—at least compared to most complex non-linear systems. These properties are the result of various dynamical constraints that are universal to conventional computers, such as the clock mechanism that synchronises the update of logic gates and other components; the ubiquitous discretization steps (where continuous values are discretized into binary 1s and 0s); and the almost complete isolation of internal processes of computers from the environment of the computer. We are investigating an alternative computational medium composed of signalling synthetic protocells to explore the implications of relaxing some of these dynamical constraints that are typical of conventional computers. Is it possible to build useful and/or programmable computers out of unconventional media such as protocells that do not have a synchronizing clock? Or that do not employ a conventional representation of 0s and 1s? Or that are less decoupled from their environment?

The protocells that we are investigating are aqueous droplets suspended in oil. Each droplet contains the reagents for the Belousov-Zhabotinsky (BZ) oscillating chemical reaction (Zhabotinsky, 2007), resulting in self-exciting dynamical units that, when in contact with each other, are capable of propagating signals similar in some respects to signal transduction in biological neurons. Networks of these signalling protocells are therefore a kind of *wet* artificial neural network, sharing more in common with biological nervous tissue than conventional computer electronics.

It is envisaged that in the future more advanced protocells will be employed to make self-organising computers, or computers that can operate within the human body. But first it is necessary to develop a better understanding of how complex non-linear systems can be harnessed to accomplish useful or “minimally-cognitive” tasks (Beer, 2003) such as categorical perception, boolean logic, and dynamical control.

Moreover, by learning how to construct or assemble networks of complex non-linear units like the BZ-protocells

we also gain insight into how other complex and non-linear “computational” media (such as nervous tissue) can conduct, modify and modulate signals and information, and how it can play an important role in the sensorimotor loops of a situated and embodied agent (Stewart et al., 2011). This bottom-up approach to the construction of alternative computational media is an important complement to the more widespread top-down neuroscience where biological neural networks are slowly being reverse engineered.

With these long and medium-term goals in mind, we have set out to (i) design functional collections of signalling protocells (comparable to the logic gates of conventional computing) that could be combined to produce more complex networks, (ii) identify effective signal encoding(s) that facilitate the transmission and manipulation of the signal by protocell networks, and (iii) identify design techniques and methodologies for creating functional signalling protocell networks out of complex non-linear media. To accomplish these goals, we are taking a three pronged approach involving *in vitro* experimentation, simulation and modelling to investigate the dynamical properties of the protocells and networks thereof; and experimental computer-aided design and machine-learning techniques to partially automate the development of functional protocell networks. We now briefly summarize our published results, before describing our current efforts.

## Summary of published research

To elucidate the experimental foundations of working with wet chemical computers on microfluidic chips (King et al., 2012), the NeuNeu project consortium ([www.neu-n.eu](http://www.neu-n.eu)) has conducted various research projects involving simulation, modelling and experimentation. One branch of this research involves the investigation of droplet networks, where the droplets are assumed to be small enough that internal spatial dynamics can be ignored. In this vein, the computing potential of two-droplet systems has been demonstrated in experiment and simulation (Szymanski et al., 2011) and differential equation models have been identified that allow us to accurately describe droplet dynamics and interactions (Szy-



manski et al., 2011). More abstract simulation models have also been developed to make possible faster and larger-scale simulations (Gruenert et al., 2013), allowing us to analyse higher-level design principles and questions pertaining to system architecture, such as possible benefits of moving beyond naive or simple signal encodings (e. g. high firing-rate = 1, and low firing-rate = 0) to explore various alternatives (Gruenert et al., 2012).

In a parallel branch of simulation and experimental work, our collaborators have been investigating more spatial forms of computing, involving larger reservoirs containing sub-excitable BZ medium. In these conditions, isolated spatial propagating waves can form, combine and interfere in spatial and geometrical ways to accomplish computation-like tasks, such as logic gates (Holley et al., 2011; Adamatzky et al., 2012).

### Ongoing research

#### Information measures for analysing and guiding the artificial evolution of unconventional computational media.

Following information theory (Shannon and Weaver, 1948) and information dynamics measures (Lizier, 2013) in cellular automata and in neural networks (Vicente et al., 2011), which help to identify information propagation, storage and modification systems, we are developing analysis tools for understanding the information flows of experimental and simulated droplet systems. These tools are intended to aid in the tracking and understanding of the flow of information through unconventional computational media, in a way that is largely independent of the encoding of the information and to thereby facilitate the search for complex and potentially useful system behaviours in random or evolved droplet networks, which are inherently less modular and decomposable than conventional engineered computational systems. We are also exploring the use of information theoretical measurements to constrain the design of functional networks. By identifying necessary changes in the state of information at different stages of computation, we believe it may be possible to guide machine-learning algorithms to more effectively design functional networks.

#### Defining computational-unit fitness implicitly using tautological closed loops.

To facilitate the artificial evolution of Basic Composable Units (“BCUs” – c.f. logic gates) for unconventional computational media, we are developing a novel technique in which optimal BCU behaviour is defined not explicitly (“given this input, the unit should produce that output”), but implicitly, through its influence on network properties in a closed network consisting of multiple instances of the unit. The network is designed in such a way that only if the units are performing the desired task (e. g. acting as a NAND gate), will certain network properties hold (e. g. dynamics at

two points in the network should be similar to each other and different to a third point), and machine-learning techniques tune the BCU properties to maximise these network properties. In this way, we implicitly describe the desired behaviour of the units without overly constraining their design, allowing the artificial evolution to concurrently design the BCUs and the encoding of the signal that they operate upon.

**Acknowledgements.** The research was supported by the NEUNEU project (248992) ([www.neu-n.eu](http://www.neu-n.eu)) sponsored by the European Community within FP7-ICT-2009-4 ICT-4-8.3 - FET Proactive 3: Bio-chemistry-based Information Technology (CHEM-IT) program.

### References

- Adamatzky, A., Holley, J., Dittrich, P., Gorecki, J., Costello, B. D. L., Zauner, K.-P., and Bull, L. (2012). On architectures of circuits implemented in simulated belousov–zhabotinsky droplets. *Biosystems*, 109(1):72–77.
- Beer, R. D. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11(4):209–243.
- Gruenert, G., Escuela, G., and Dittrich, P. (2012). Symbol representations in evolving droplet computers. In Durand-Lose, J. and Jonoska, N., editors, *Proc. of Unconventional Computation and Natural Computation - 11th Intl. Conf. Orléan, France, September 3-7, 2012.*, volume 7445 of *Lecture Notes in Computer Science*, pages 130–140. Springer.
- Gruenert, G., Szymanski, J., Holley, J., Escuela, G., Diem, A., Ibrahim, B., Adamatzky, A., Gorecki, J., and Dittrich, P. (2013). Multi-scale modelling of computers made from excitable chemical droplets. *International Journal of Unconventional Computing*, 9(3–4):237–266.
- Holley, J., Jahan, I., Costello, B., Bull, L., and Adamatzky, A. (2011). Logical and arithmetic circuits in Belousov Zhabotinsky encapsulated discs. *Physical Review E*, 84(5):056110.
- King, P. H., Corsi, J. C., Pan, B.-H., Morgan, H., de Planque, M. R., and Zauner, K.-P. (2012). Towards molecular computing: Co-development of microfluidic devices and chemical reaction media. *Biosystems*, 109(1):18–23.
- Lizier, J. T. (2013). *The local information dynamics of distributed computation in complex systems*. Springer.
- Shannon, C. E. and Weaver, W. (1948). A mathematical theory of communication. *The Bell Systems Technical Journal*, 27:379–423, 623–656.
- Stewart, J., Gapenne, O., and Di Paolo, E. A. (2011). *Enaction: Toward a New Paradigm for Cognitive Science*. MIT Press.
- Szymanski, J., Gorecka, J. N., Igarashi, Y., Gizynski, K., Gorecki, J., Zauner, K.-P., and Planque, M. D. (2011). Droplets with information processing ability. *International Journal of Unconventional Computing*, 7(3):185–200.
- Vicente, R., Wibrall, M., Lindner, M., and Pipa, G. (2011). Transfer entropy: a model-free measure of effective connectivity for the neurosciences. *Journal of computational neuroscience*, 30(1):45–67.
- Zhabotinsky, A. M. (2007). Belousov-zhabotinsky reaction. *Scholarpedia*, 2(9):1435.



## Heterogeneity and complexity of a simulated terrestrial environment account for superiority of the altruistic gene

Tadao Maekawa<sup>1</sup>, Manabu Honda<sup>2</sup>, Norie Kawai<sup>3</sup>, Emi Nishina<sup>4</sup>, Osamu Ueno<sup>2</sup> and Tsutomu Oohashi<sup>3</sup>

<sup>1</sup>Faculty of Environmental and Information Sciences, Yokkaichi University

<sup>2</sup>Department of Functional Brain Research, National Center of Neurology and Psychiatry

<sup>3</sup>Department of Research and Development, Foundation for Advancement of International Science

<sup>4</sup>The Open University of Japan / The Graduate University for Advanced Studies

### Abstract

Recent research on the notion of altruism in terrestrial life has focused on certain altruistic behaviors, which are regarded as beneficial to animal life, especially with respect to individual animal species. Such findings throw light on individual-oriented mechanisms and their evolution in helping to clarify so-called intentional interactions between individuals based on discrimination of other individuals and remembered information as advanced by developments in biological information processing, ranging from molecular recognition to activation of the neural system. In 2006, Nowak classified these mechanisms into five types. In the current study, we have zeroed in on the process of autolysis universally observed in all terrestrial lives, as characterized by genetically programmed death accompanied by altruistic self-decomposition, whose model we call the “programmed self-decomposition model (PSD Model)”. In our view, altruistic phenomena target no specific individuals yet prove beneficial to the ecosystem, in part and as a whole. Using our PSD Model we ran evolutionary simulations of altruistic phenomena in the SIVA Series, which is an artificial life system designed to resemble a terrestrial ecosystem, and one that excludes both discrimination of individuals and interactions between individuals. In our simulations no individual-oriented evolutionary mechanism was observable while the ecosystem-oriented mechanism positively contributed to the evolution of the altruistic gene. Our research has thus sought to determine factors that promote superior evolutionary characteristics of altruistic phenomena in a terrestrial ecosystem model. The current study argues that the high heterogeneity and complexity of a terrestrial environment and the eternality of evolutionary time play an important role in the selective process of programmed death in the terrestrial ecosystem, which is accompanied by altruistic self-decomposition. Based on the above findings, we investigated the inseparable relationship existing between a terrestrial ecosystem and the altruistic gene.

### Introduction

We previously modeled autonomous death that comprises a universal attribute of terrestrial life, as programmed self-decomposition (PSD) (Oohashi et al. 1987, 2009). Our research centers on a series of studies that delve into the existence of autonomous death through experiments in the field of molecular cell biology using existing living organisms as subjects; concurrently, in carrying out evolutionary simulations of Artificial Life (ALife), we raise the possibility

that mortal organisms having autonomous death are superior to immortal organisms (Oohashi et al. 1987, 1999, 2001, 2009, 2011, 2014; Maekawa et al. 2011). The essence of our PSD model zeroes in on the process of autolysis (Odum 1971), which is universally observed in terrestrial lives including unicellular organisms, as phenomena with respect to recycling of autonomous material in a terrestrial ecosystem. Conventionally autolysis has been regarded as deregulated, natural disintegration with increasing entropy. We have redefined autolysis as a type of autonomous, altruistic phenomenon beneficial to an ecosystem, in part or as a whole. We thus regard autolysis as an active biochemical process built into cellular genetic programming by which a cell consumes its own metabolic energy. In view of this autolytic process, we posit that life individuals autonomously decompose themselves into components; in other words, cells hydrolyze biological polymers into biological monomers so that the materials they consume and the spaces where they exist can be optimally reutilized by all other life individuals, including adversaries and competitors, and, by means of that event, can thereby return to the environment and thus contribute to the restoration of the entire ecosystem.

Recent research on the concept of altruism in terrestrial life has focused on certain altruistic behaviors regarded as beneficial to animal life, especially with respect to individual animal species (Haldane 1932; Hamilton 1963; Price 1970). Based on these many previous researches, Martin A. Nowak has provided a useful framework that classifies the mechanisms of evolution of cooperation under five types (Nowak 2005, 2006, 2011, 2012). His rules quite adequately account for altruistic phenomena that targets only specific individuals or groups. Nowak’s five rules for these mechanisms require, as prerequisite functions, discrimination of other individuals and reference to remembered information, as advanced by developments in biological information processing, ranging from molecular recognition to activation of the neural system. Such altruistic behavior is realized by the individual-oriented mechanisms whose actual property is the intentional interaction between individuals based on such rules. The terrestrial lives that Nowak’s framework of altruistic behavior encompasses are limited to relatively evolved animals that deploy biological control systems that enable the discrimination between individuals and the remembering of an individual’s experience (Oohashi et al.

2014), at least as chemical messengers, and, ideally, a central nervous system sufficiently robust for formation intention.

We have redefined an altruistic phenomenon as being a phenomenon by which a life individual renders certain biological benefits to a part of the ecosystem including individuals as well as to the ecosystem as a whole, regardless of any biological benefit for or disadvantage to itself (Oohashi et al. 2011, 2014). In our view, the recipient of a contribution should not only be limited to a specific individual or a group of individuals; but rather, include the ecosystem, in part or as a whole. Consequently, we put forth the programmed self-decomposition model as an ecosystem-oriented altruistic phenomenon that emerges even for very primitive life individuals equipped with only the fundamental principle of terrestrial life, namely, self-reproduction and self-decomposition regulated solely by a genetic program, and without any functionality by which to discriminate between individuals (Oohashi et al. 2014). Considering that the essential quality of the PSD consists of autolysis and that organelles (lysosome) executing PSD exist in every eukaryote cell, it would be possible for PSD to serve as a mechanism universally existing in terrestrial lives, and for it to exist as a universal basic mechanism for all eukaryotes, including animals that produce the individual-oriented altruistic behavior proposed by Nowak.

Based on the conceptual clarification of altruistic phenomena, we constructed a simulator system SIVA Series equipped with primitive artificial life in an ecosystem designed to resemble a terrestrial ecosystem, and one that excludes both discrimination of individuals and interactions between individuals. Through a series of simulation studies, we showed many conditions whereby evolutionary adaptation is promoted by means of altruism even in extremely primitive lives equipped only with the basic principle of terrestrial life, that is, the self-reproduction and self-decomposition regulated solely by a genetic program (Oohashi et al. 1987, 1999, 2001, 2009, 2011, 2014; Maekawa et al. 2011). Especially noteworthy is our finding that the gene of altruistic death accompanied by programmed self-decomposition can be acquired through the evolution of immortal lives, and the lives that acquire the gene of altruistic death sometimes are overwhelmingly superior to immortal lives (Oohashi et al. 2014, Maekawa et al. 2011). Accordingly, we regard altruistic death accompanied by programmed self-decomposition as a sophisticated survival strategy acquired as the fruit of evolution. We thus categorize lives that have completed this

evolution as altruistic mortal lives and more primitive lives that have yet to complete such evolution as non-altruistic immortal lives.

This study examined the possible reasons why altruistic phenomena of programmed self-decomposition brought forth evolutionary superiority in terrestrial ecosystem models. We therefore hypothesized that the high heterogeneity and complexity of terrestrial environments and the eternity of evolutionary time of a terrestrial ecosystem played important roles in the selective process of programmed death, which is accompanied by altruistic self-decomposition in terrestrial ecosystems. In our experimental models using artificial lives, we carried out experiments that tested our hypothesis and obtained positive results. Here we describe the results.

## Methods

### 1) Design of the SIVA simulator and its virtual environment

In the present study, we again used SIVA-T05 as an evolution simulator. Its construction and functions are the same as those utilized in a previous report (Oohashi et al. 2009).

To simulate the characteristics of a terrestrial environment using a limited amount of materials distributed in a finite space, the virtual space of SIVA-T05 is designed to be a two-dimensional lattice consisting of  $16 \times 16$  (= 256) spatial blocks. A single spatial block is defined as  $8 \times 8$  (= 64) pixels for habitation points. One habitation point is occupied by one virtual life individual (VLI) and vice versa (Figure 1a). Environmental conditions can be independently defined for each spatial block and those of the 64 habitation points in the same spatial block are configured to always be homogeneous. Since all VLIs in one spatial block share identical environmental conditions, the population of VLIs in that block significantly affects local conditions.

In the present study, the temperature gradient and the initial distribution of four kinds of virtual inorganic biomaterials making up the VLIs were set to be heterogeneous or homogenous across the whole ecosystem according to experimental conditions (see next section). No substances other than virtual inorganic biomaterials existed in the initial environment.

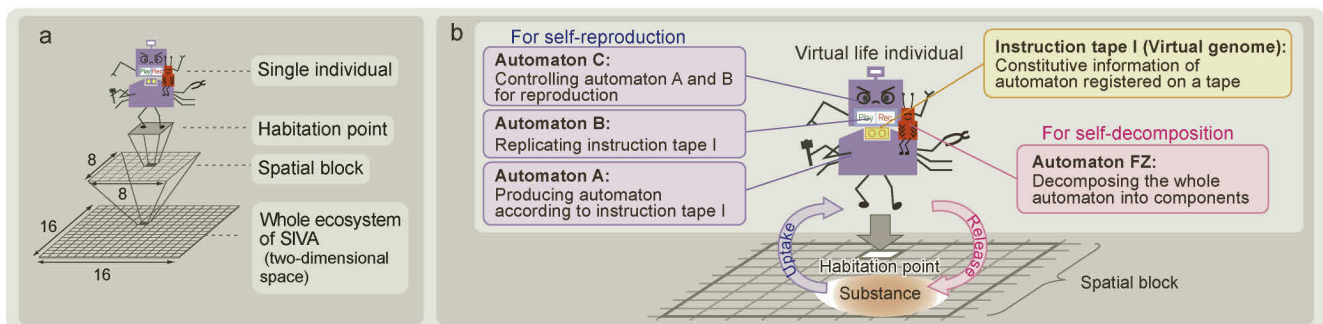


Figure 1. Environmental design and life activities of virtual life individuals (VLIs) of the virtual ecosystem SIVA-T05. a) Environmental design. b) Relationship between life activities of virtual life individuals (VLIs) and the environment.

## 2) Structure and behavior of artificial life

**Structure of a virtual life individual.** As in the previous report (Oohashi et al. 2009), we designed a virtual life individual (VLI) based on Oohashi's self-reproductive, self-decomposable (SRSD) automaton model (Figure 1b) that took von Neumann's self-reproductive automaton model (Von Neumann 1951) as its prototype (Figure 2). Oohashi's automaton G is described as  $G = D + FZ + I_{D+FZ}$ , where  $D = A + B + C$ . Here, automaton A produces automata according to instructions on data tape I (that is, a virtual genome). Automaton B reads and replicates data tape I. Automaton C sets the copy of data tape I replicated by automaton B into new automata produced by automaton A and separates these as automaton D. Automaton FZ, which is a modular subsystem plugged into automaton D, decomposes the whole automaton G into components suitable for reutilization when automaton G encounters serious environmental conditions in which it is unable to live or has reached the end of its life span. Data tape  $I_{D+FZ}$  carries an instruction describing automaton D + FZ. Thus, automaton G, which corresponds to  $D + FZ + I_{D+FZ}$ , can reproduce an identical automaton G as well as decompose itself.

We designed artificial life based on AChem (Dittrich et al. 2001; Suzuki 2004) so as to realize the above-mentioned

logical behaviors and, as faithfully as possible, to reflect the principles of terrestrial life and its subsequent reproduction. We constructed a VLI from four classes of virtual biomolecules: virtual inorganic biomaterials (VI), that are four kinds of substances distributed in the environment; virtual organic biomaterials (VO); virtual biological monomers (VM); and, virtual biological polymers (VP). Any molecules in the latter three classes consist of combinations of the four kinds of VI. A virtual genome in the VP class consists of virtual nucleotides belonging to the VM class. The virtual protein in the VP class is produced according to a sequence of virtual nucleotides that determines the primary sequence of virtual amino acids belonging to the VM class (Oohashi et al. 2009). We developed a SIVA language that actualizes virtual life activities by recognizing the sequence of the virtual amino acids contained in the virtual protein as coded program sentences then executes the specific life activity. According to given conditions, this SIVA language reproduces, divides, and decomposes a VLI.

Each VLI expresses its life activities by executing all sentences satisfying their execution conditions in the VLI during one Time Count (TC), the unit of virtual time in SIVA-T05. The order for a VLI in the virtual ecosystem to express its life activities within one TC is randomly determined at every TC. It takes at least 5 TCs for a newborn individual to reproduce itself in our current simulation experiments.

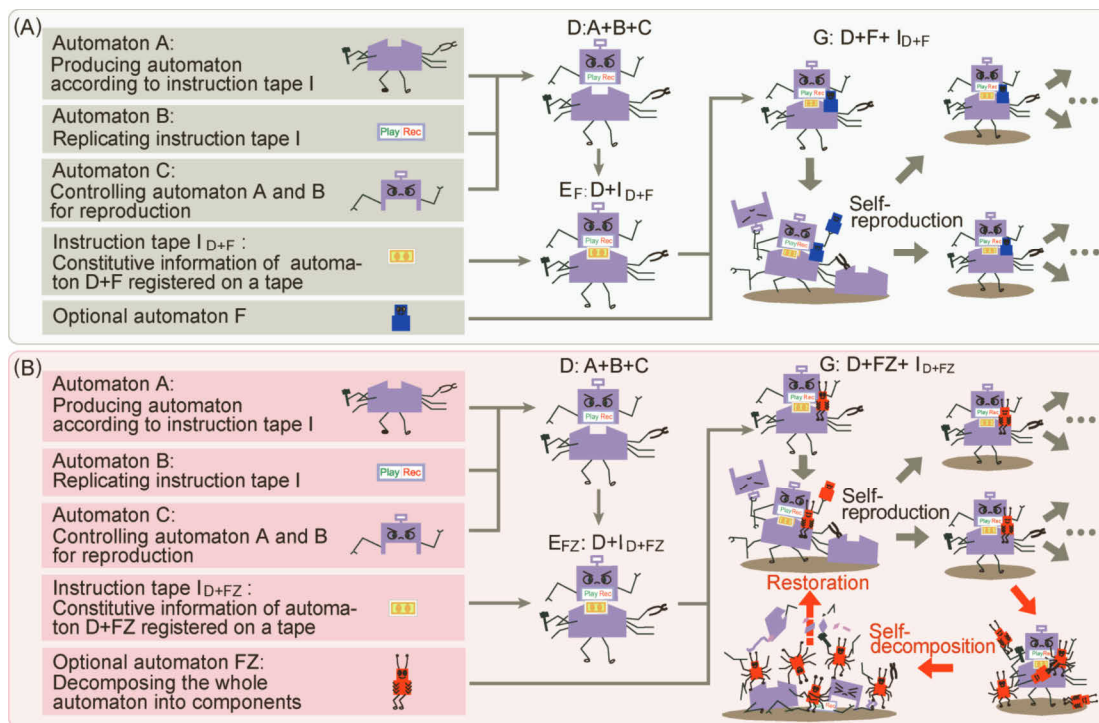


Figure 2. Von Neumann's self-reproductive automaton and Oohashi's self-reproductive, self-decomposable automaton.

(A) Von Neumann's self-reproductive automaton model. This is an immortal type model without an autonomous mechanism for the restoration of the environment to its original state. (B) Oohashi's self-reproductive, self-decomposable (SRSD) automaton model. This model uses von Neumann's self-reproductive automaton model as its prototype. It has a programmed mechanism contributing to the restoration of the environment to its original state through autonomous individual death with self-decomposition, which is an essential feature of terrestrial life. Two activation modes are defined for the self-decomposition automaton FZ. The first one is activated by a signal input from outside, indicating unconformity between the life and its habitation environment. The second mode constitutes the end of the life span.



Therefore, we use “passage duration” as a virtual time unit, which corresponds to the value of TC divided by 5 (Oohashi et al. 2009).

**Behavior of virtual life individuals.** A VLI executes its life activities by consuming materials from the virtual environment (Oohashi et al. 2009). Activities of each VLI are so designed as to depend on the amount of material available as well as the temperature in the inhabited spatial block. Namely, optimum environmental conditions are defined for each VLI a priori. Activities of a VLI decrease when environmental conditions of the habitation point move away from VLI optimum points. A VLI cannot express its life activities when environmental conditions markedly deviate from the optimum, and, in the case of a mortal organism, it decomposes itself just as it does when it has lived out its life span. Materials released by the decomposition of a VLI are restored to the environment and become utilizable by other individuals as well as the space that were occupied by the VLI.

When VLIs reproduce, point mutation can occur at a predefined probability during replication of the virtual genome. Mutations may alter the optimum environmental conditions for a VLI such as temperature and the composition of the VIs. In addition, VLIs which can use only VI for self-reproduction, which requires a greater amount of energy, can evolutionarily become those which can also use VM, which requires smaller amount of energy. In other words, the evolution of the material uptake function is also installed. These mutations enable the VLI to live in an environment where it originally could not live. That is to say, evolutionary adaptation to the environment can occur.

### 3) Experimental setting

To evaluate the hypothesis that the high heterogeneity of terrestrial environments and the eternity of the evolutionary time of a terrestrial ecosystem played an important role in the selective process of programmed death, which is accompanied by altruistic self-decomposition of the terrestrial ecosystem, we employed artificial life in three experimental conditions in which different initial distributions of VIs and temperature were employed as shown in Figure 3: experimental condition A: initial distribution of both VIs and temperature optimal level for VLIs was homogenous throughout the whole environment; experimental condition B: initial distribution of

VLIs was homogenous whereas initial distribution of temperature was heterogeneous; and, experimental condition C: initial distributions of both VIs and temperature were heterogeneous. Under each experimental condition, we seeded an altruistic mortal VLI and a non-altruistic immortal VLI in spatial blocks in the midst of the simulation space whose environmental conditions were most suitable for these VLIs to start simulations of their reproduction and evolution. We conducted 100 simulations of 800 passage durations with a mutation rate of 0.005 and observed changes in the size of habitation area and number of VLIs. Since mutation occurs at each reproduction according to the configured mutation rate under the current experimental conditions, we calculate the approximate magnitude of the mutations that occur during the simulation by means of the total number of reproductions. Therefore, we aggregated the number of reproductions for both mortal VLIs and immortal VLIs to compile the cumulative mutation index.

## Results

Figure 4 shows the typical transition pattern of distribution of VLIs, their number, and the cumulative mutation index of more than 800 passage durations for each of three conditions set according to the level of environmental heterogeneity. Table 1 shows the ratio of survival of VLIs up to either the 400<sup>th</sup> passage duration or the 800<sup>th</sup> passage duration, average and standard deviation ratio of number of mortal VLIs to that of immortal VLIs at the 400<sup>th</sup> and at the 800<sup>th</sup> passage duration when immortal VLIs survived, and the cumulative mutation index for both mortal VLIs and immortal VLIs at the 800<sup>th</sup> passage duration of a typical example as seen in Figure 4.

Under Condition A, when both substances and temperature were homogeneous, immortal VLIs were greater in number with continued reproduction but after the 200<sup>th</sup> passage duration, the VLIs filled the entire simulation space and entered a stable phase. On the other hand, mortal VLIs reproduced themselves until the 10<sup>th</sup> passage duration after the onset of simulation. However, when passage duration exceeded 10, reproduction of mortal VLIs stagnated that reduced their number so that there were twice as many immortal VLIs as mortal VLIs at the 15<sup>th</sup> passage duration. After 20 passage durations, the number of mortal VLIs once again increased at a rate of increase similar to that of immortal

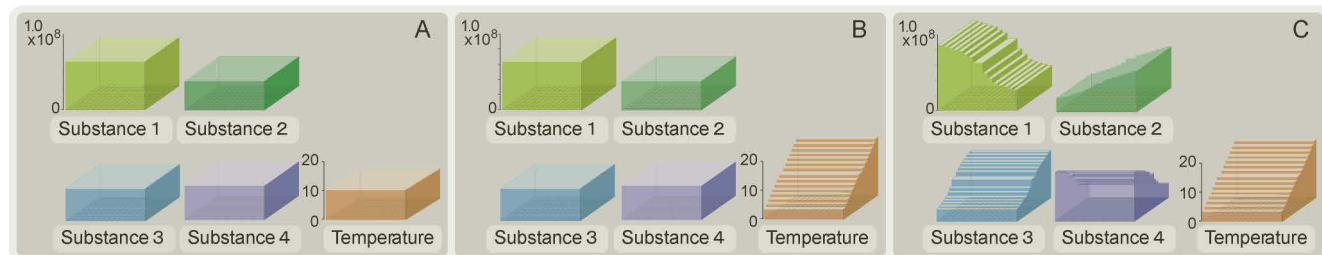


Figure 3. Three experimental conditions showing initial distribution pattern of virtual inorganic biomaterials (VIs) and temperature.

A) VIs: homogeneous, Temperature: homogeneous. B) VIs: homogeneous, Temperature: heterogeneous. C) VIs: heterogeneous, Temperature: heterogeneous.

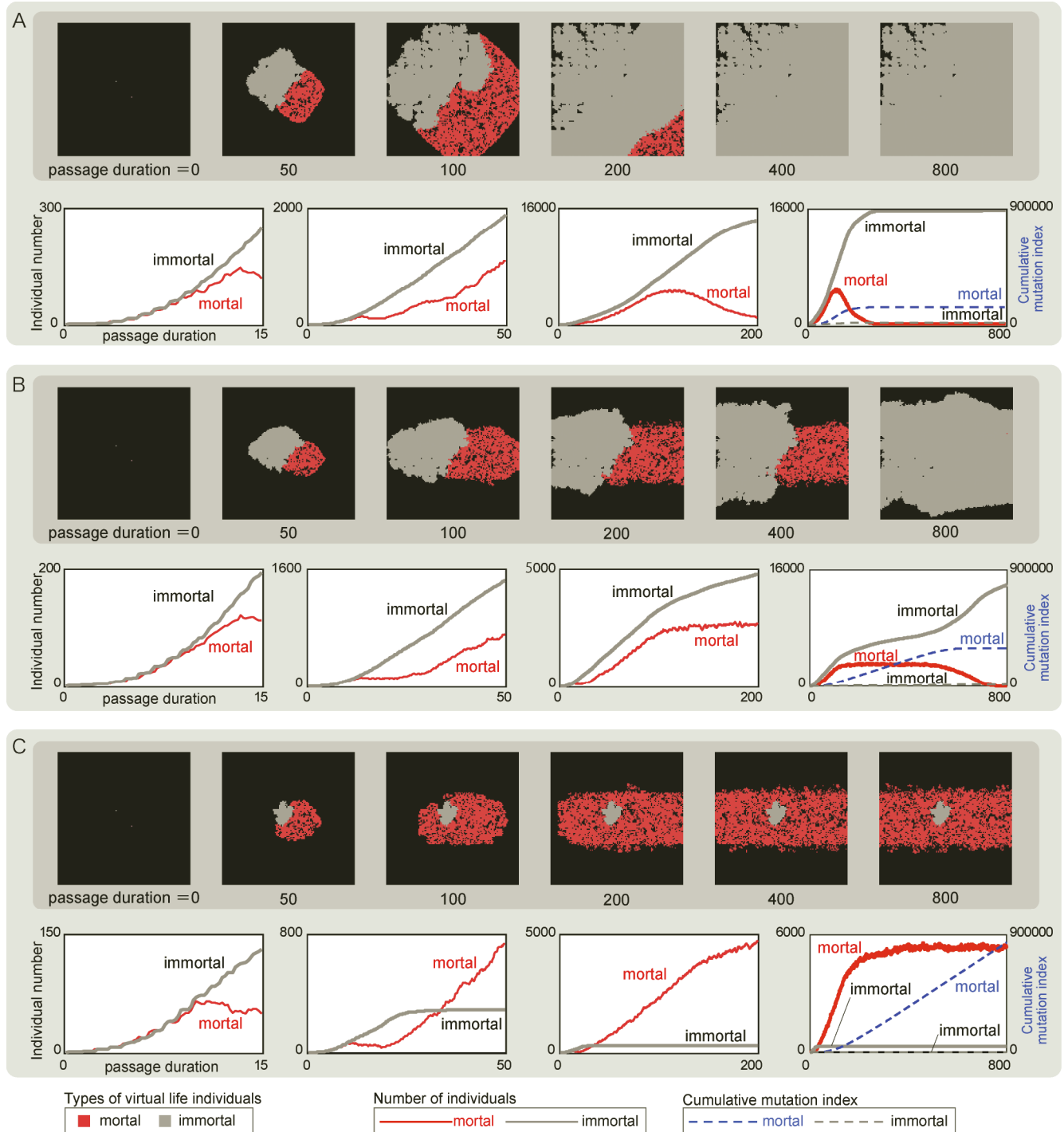


Figure 4. Altruistic gene became more superior as heterogeneity and complexity of the environment became greater. Successive changes in individual distribution (upper panels), the number of individuals (solid lines in lower panels), and the cumulative mutation index (dotted lines in the lower right panel) of mortal and immortal virtual life individuals (VLIs) simulated under each experimental condition: A) VIs: homogeneous, Temperature: homogeneous. B) VIs: homogeneous, Temperature: heterogeneous. C) VIs: heterogeneous, Temperature: heterogeneous.



Table 1: The superiority of the mortal VLIs led by the increase in heterogeneity and complexity in environmental conditions.

Condition			400 <sup>th</sup> passage duration		800 <sup>th</sup> passage duration		Cumulative mutation index		
			Ratio of survival (mortal VLs)	Average and SD ratio of number (mortal VLs / immortal VLs)	Ratio of survival (mortal VLs)	Average and SD ratio of number (mortal VLs / immortal VLs)	Typical values	Ratio (mortal VLs / immortal VLs)	
Substance (VLs)	Temperature								
A	homo-geneous	homo-geneous	0%	0	0%	0	mortal immortal	133910 15910	8.4
B	homo-geneous	hetero-geneous	8%	0.19±0.17	0%	0	mortal immortal	294747 13100	22
C	hetero-geneous	hetero-geneous	11%	18±2.1	11%	18±2.0	mortal immortal	840945 291	2890

VLIs. However, mortal VLIs increased until the 120<sup>th</sup> passage duration at which time they entered a reduction phase and died out after the 200<sup>th</sup> passage duration. Such a tendency was commonly observed in all of the 100 simulations executed, and mortal VLIs never survived until the 400<sup>th</sup> passage duration in any of the simulations. However, while the cumulative mutation index at the 800<sup>th</sup> passage duration of immortal VLIs was 15910, that of mortal VLIs reached 133910, 8.4 times that of the immortal VLIs. That finding shows that the magnitude of the cumulative mutation of mortal VLIs was much higher than that of immortal VLIs.

Under Condition B, when the temperature was heterogeneous while the initial distribution of substances was homogeneously distributed, immortal VLIs steadily increased and maintained a greater number than mortal VLIs. After reaching the 100<sup>th</sup> passage duration, the increase of the immortal VLIs slowed down then accelerated once again after the 600<sup>th</sup> passage duration in parallel with the decrease of mortal VLIs. In contrast, mortal VLIs started to reproduce but its number deceased at around the 10<sup>th</sup> passage duration. Then, at the 20<sup>th</sup> passage duration, mortal VLIs increased once again until reaching a plateau at around the 120<sup>th</sup> passage duration. At the same time, mortal VLIs remained constant in number and well-balanced with respect to reproduction and decomposition up to the 500<sup>th</sup> passage duration, and began to decrease at the 600<sup>th</sup> passage duration then dying out at the 800<sup>th</sup> passage duration. The ratio of survival rate of mortal VLIs at the 400<sup>th</sup> passage duration was 8%, but the number of mortal VLIs was, on average, 0.19 times that of immortal VLIs, which shows the absolute superiority of immortal VLIs to mortal ones in all trials. Furthermore, when we extended the evolutionary time of the simulation, mortal VLIs died out before reaching the 800<sup>th</sup> passage duration without exception. However, while the cumulative mutation index at the 800<sup>th</sup> passage duration of immortal VLIs was 13100, that of the mortal VLIs was 294747, which reached 22 times that of immortal VLIs. This finding shows that the magnitude of the cumulative mutation of mortal VLIs was even higher than that of immortal VLIs as compared to Condition A.

Under Condition C, when both temperature and initial distribution of substance were heterogeneous, immortal VLIs began reproduction in the same way as they did in Condition A and Condition B, but ceased reproduction after the 25<sup>th</sup>

passage duration, and were completely surpassed by mortal VLIs at the 30<sup>th</sup> passage duration. There was no notable change observed either in number or size of the area in which the immortal VLIs existed until they reached the 800<sup>th</sup> passage duration. On the other hand, mortal VLIs started to reproduce but its number deceased at around the 10<sup>th</sup> passage duration. Then, at the 20<sup>th</sup> passage duration, mortal VLIs increased once again and surpassed immortal VLIs in number at around the 30<sup>th</sup> passage duration, and continued to increase in number and size of the area in which they existed. At the 200<sup>th</sup> passage duration, the rate of increase declined but maintaining a stable balance between reproduction and decomposition up to the 800<sup>th</sup> passage duration. The number of trials in which mortal lives survived until the 400<sup>th</sup> passage duration increased from 8 to 11 out of 100 trials. They survived until the 800<sup>th</sup> passage duration. Both at the 400<sup>th</sup> and 800<sup>th</sup> passage duration, the number of mortal VLIs was, on average, 18 times greater than the number of immortal VLIs, which shows the overwhelming prosperity of the former. While the cumulative mutation index at the 800<sup>th</sup> passage duration of immortal VLIs was 291, that of mortal VLIs was 840945 (2890 times). This shows that the greatest magnitude of cumulative mutation was accumulated in mortal VLIs among the three environmental conditions with respect to heterogeneity. As shown above, when an environment is more heterogeneous and complex, the survival rate of mortal VLIs increases and the duration of survival becomes longer so that mortal VLIs overwhelm immortal VLIs even in the number of individuals in a heterogeneous and complex condition. The cumulative mutation was observed to be markedly greater for mortal VLIs than for immortal ones. Furthermore, it is noteworthy that even when mortal VLIs overwhelmed immortal ones in the final stage under a heterogeneous and complex experimental environment, immortal VLIs dominated mortal ones at the initial stage, up to the 30<sup>th</sup> passage duration without exception. This indicates that a certain length of time for evolution and prosperity is necessary before mortal VLIs can surpass and overwhelm immortal ones.

## Discussion

Using the SIVA Series, an artificial life system designed to resemble a terrestrial ecosystem that excludes both

discrimination of individuals and interactions between individuals, we examined factors that promote superior evolutionary characteristics of altruistic phenomena and a gene having altruistic properties.

Our results showed that immortal lives without altruistic properties are overwhelmingly more prosperous than altruistic mortal lives in a more homogeneous and simple artificial ecosystem with a shorter reproductive evolutionary time. On the other hand, the more heterogeneous and complex an environment having a longer reproductive evolutionary time is, the more dominant the altruistic mortal lives become over non-altruistic immortal lives. That is, when the substances necessary for reproduction of individual life and the temperature necessary for emergence of life-form activities were homogeneously distributed under simple environmental conditions at the initial stage, altruistic mortal lives were overwhelmed by non-altruistic immortal lives and never survived up to 400<sup>th</sup> passage duration in any of 100 trials. When, at the initial stage, the temperature was heterogeneous while the substances were homogeneously distributed as above, mortal lives survived up to the 400<sup>th</sup> passage duration in 8 out of 100 trials. Note, however, that the number of immortal lives exceeded the number of mortal lives at the 400<sup>th</sup> passage duration in all those 8 trials. Furthermore, when we extended the evolutionary time of the simulation, the mortal lives died out before reaching the 800<sup>th</sup> passage duration in all 8 trials.

On the other hand, when the temperature was heterogeneous and the initial distribution of substances necessary for living organisms was heterogeneously distributed to increase the complexity of environmental conditions, the number of trials in which mortal lives survived until the 400<sup>th</sup> passage duration increased from 8 to 11 out of 100 trials. Noteworthy, in all of the 11 trials in which mortal lives survived, mortal lives were clearly inferior to immortal lives initially but the situation eventually reversed, and, at the 400<sup>th</sup> passage duration, the number of mortal lives was, on average, 18 times greater than those of immortal lives; i.e., the mortal lives overwhelmingly prospered. These results show that an increase in heterogeneity and complexity of environmental conditions significantly improved the superiority of mortal lives with respect to immortal lives.

From the temporal aspect, it was observed that, at the initial stage, under initial conditions either homogeneous or heterogeneous, immortal lives were dominant over mortal lives in all cases, and, only when mortal lives were able to escape from extinction did they overwhelmingly dominate immortal lives without exception, although a long evolutionary time span was required. Such results indicate that if a long reproductive evolutionary time span accompanies high heterogeneity and complexity of environment, the altruistic mortal lives will become superior with respect to non-altruistic immortal lives.

These results support our hypothesis that given a sufficiently long evolutionary time span, the high heterogeneity and complexity of the terrestrial environment plays an important role in the evolutionary selection of the gene with programmed death accompanied by altruistic self-decomposition in the terrestrial ecosystem.

Life forms accustomed to existing in an optimal environment in a heterogeneous ecosystem have a greater chance of

encountering an unconformable environment as these life forms continue to reproduce and increase the size of the area in which they exist.

An unconformable environment is nothing but an environment in which reproduction is made difficult or impossible. To survive therein, such life forms must undergo evolutionary adaptation, thus acquiring novel life activity that is amenable to such environmental conditions. Immortal lives in such an environment would have no further chance to produce new individuals when the area possible for reproduction is completely filled. Therefore, before an area appropriate for reproduction fills up, there must be a mutation that provides evolutionary adaptation enabling survival in an adjacent area. Otherwise, both reproduction and evolution are blocked. As the heterogeneity in the environment increases, or as the areas with survivable homogeneous environmental conditions decrease, reproduction becomes difficult for an immortal life and the possibility of evolutionary occlusion increases.

On the other hand, mortal lives can continue the alternation of generation, even within a small area, by returning substances and space to the environment through self-decomposition and recycling. Therefore, mortal lives always possess the potential to achieve novel evolutionary adaptation by accumulating mutations without falling into the blockage of evolution. Alteration in characteristics by mutation in this case emerges as a change in the balance of the inorganic substances necessary for reproduction, an acquisition of the function of monomer intake, and a shift in optimal temperature. Thus, as the heterogeneity and complexity of environmental conditions increase, the activities of immortal lives decrease, and, at the same time, mortal lives attain superiority.

In simulations in which the initial conditions of the environment were set to be highly heterogeneous, the mortal lives overwhelmingly prospered. However, even in such cases, immortal lives were, as is natural, dominant at the initial stage of the simulation without exception. It was noteworthy that the situation completely reversed later and altruistic mortal lives that had been weak became superior.

The temporal pattern of the number of life individuals along the time line of reproductive evolution displayed particular characteristics. Under all conditions, mortal lives smoothly began reproduction and increased until the 10<sup>th</sup> passage duration at which time they entered a reduction phase. The number of mortal lives remained small for a time, then, prior to the 30<sup>th</sup> passage duration, reverted to the increase phase. When the environmental conditions became heterogeneous, the number of mortal lives monotonically increased, overwhelming the number of immortal lives.

Mutations accumulate during the alternations of generation. These mutations stochastically occur in all directions and do not necessarily acquire an evolution appropriate to a particular time and place. The environmental conditions to which the life forms must adapt are continuously changing. Therefore, a longer period, that is, eternal time, is necessary for an individual to acquire the characteristics appropriate for the varying environmental conditions conducive to mortal life as to ensure superiority within the whole ecosystem.

Non-altruistic immortal lives show superiority within a shorter time span. However, when environmental conditions are heterogeneous and complex, the prevalence of immortal lives is reduced and blocked while, over a longer time span,

altruistic mortal lives show significant superiority. This finding is highly suggestive for discussing the superiority of the altruistic gene in an ecosystem resembling a terrestrial one. The actual terrestrial ecosystem has consisted of multidimensional, multifaceted heterogeneous and complex environments microscopically and macroscopically throughout the Earth's entire several-billion-year history. The simulations under study here show that, in such environments, even primitive life forms having simple altruistic mechanisms for decomposing themselves so that they can contribute to the ecosystem in part and as a whole possess evolutionary potential for producing a teeming variety of genes and characteristics. Our results further suggest that the altruistic gene with characteristics physically appropriate for a terrestrial ecosystem endowed with high complexity and eternal time can be evolutionarily selected, can prosper and, as a result, can provide the basis for the Earth's biological diversity.

Nowak's framework (2006) of altruistic phenomena quite validly explains the altruistic behavior of higher species of animals. As a basis of such a highly developed individual-oriented altruism, the ecosystem-oriented altruism mechanism without intention is universally available to all terrestrial lives and thus functions as the basic mechanism for existent terrestrial life. This suggests that the Earth's environment might well possess the optimum characteristics for selecting the altruistic gene.

## References

- Dittrich, P., J. Ziegler and W. Banzhaf (2001). "Artificial chemistries - A review." *Artificial Life* 7(3): 225-275.
- Haldane, J. B. S. (1932). *The causes of evolution*. London, Longmans.
- Hamilton, W. D. (1963). "The evolution of altruistic behavior." *The American Naturalist* 97: 354-356.
- Maekawa, T., O. Ueno, N. Kawai, E. Nishina, M. Honda and T. Oohashi (2011). Evolutionary acquisition of genetic program for death. *Advances in Artificial Life, ECAL 2011: Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*. T. Lenaerts, M. Giacobini, H. Bersini et al. Cambridge, The MIT Press: 481-486.
- Nowak, M. A. (2006). "Five Rules for the Evolution of Cooperation." *Science* 314(5805): 1560-1563.
- Nowak, M. A. (2012). "Evolving cooperation." *Journal of Theoretical Biology* 299(0): 1-8.
- Nowak, M. A. and R. Highfield (2011). *SuperCooperators : Altruism, Evolution, and Why We Need Each Other to Succeed*. New York, Free Press.
- Nowak, M. A. and K. Sigmund (2005). "Evolution of indirect reciprocity." *Nature* 437(7063): 1291-1298.
- Odum, E. P. (1971). *Fundamentals of Ecology*. Philadelphia, PA, W.B. Saunders Company.
- Oohashi, T., T. Maekawa, O. Ueno and M. Honda (2011). "The supremacy of the altruistic gene: Terrestrial life has succeeded in breaking through evolutionary deadlock." *Kagaku* (in Japanese) 81(1): 83-90.
- Oohashi, T., T. Maekawa, O. Ueno, N. Kawai, E. Nishina and M. Honda (2014). "Evolutionary acquisition of a mortal genetic program: The origin of an altruistic gene." *ECAL 2011 Special Issue of Artificial Life: in press*.
- Oohashi, T., T. Maekawa, O. Ueno, N. Kawai, E. Nishina and K. Shimohara (2001). "Artificial life based on the programmed self-decomposition model, SIVA." *Artificial Life and Robotics* 5(2): 77-87.
- Oohashi, T., T. Maekawa, O. Ueno, E. Nishina and N. Kawai (1999). Requirements for immortal ALife to exterminate mortal ALife in one finite, heterogeneous ecosystem. *The 5th European Conference on Advances in Artificial Life (ECAL'99)*, London, Springer-Verlag.
- Oohashi, T., D. Nakata, T. Kikuta and K. Murakami (1987). "Programmed self-decomposition model." *Kagakuikoron* (in Japanese) 18(2): 21-29.
- Oohashi, T., O. Ueno, T. Maekawa, N. Kawai, E. Nishina and M. Honda (2009). "An effective hierarchical model for the biomolecular covalent bond: An approach integrating artificial chemistry and an actual terrestrial life system." *Artificial Life* 15(1): 29-58.
- Price, G. R. (1970). "Selection and covariance." *Nature* 227(5257): 520-521.
- Suzuki, H. (2004). Network artificial chemistry - Molecular interaction represented by a graph. *Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9) Workshop and Tutorial*, Boston, MA.
- Traulsen, A. and M. A. Nowak (2006). "Evolution of cooperation by multilevel selection." *Proceedings of the National Academy of Sciences* 103(29): 10952-10955.
- Von Neumann, J. (1951). *The general and logical theory of automata. Cerebral mechanisms in behavior - The Hixon symposium*. L. A. Jeffress. New York, John Wiley & Sons, Inc.: 1-41.

## Beyond Life Cycles

Maja Murnik, Špela Petrič and Miha Turšič

The Cultural Centre of European Space Technologies (KSEVT), Vitanje, Slovenia  
miha@ksevt.eu

### Abstract

Human activities in outer space are producing increasing quantities of space “debris”. This well-known fact posits the question about the value and use of space technologies after their operation period has expired. Rather than calling these non-functional objects “debris”, we propose to treat them as “end-of-life allopoietic systems” with the potential of becoming autopoietic systems. In general, our utilitarian, anthropocentric, and control-oriented management of processes discourages research into emancipated, unfamiliar entities which do not (yet) appear in our ecosystems. However, outer space technology with its literal and symbolic remoteness presents an opportunity to transform utilitarian objects at their end-of-life into emancipated non-utilitarian living or life-like systems without the danger of interaction with the existing living systems of our planet. Here we outline a composite approach to the challenge.

### Composite Methodologies for Outer Space

From the beginnings of Modern age, artistic and scientific communities have been epistemologically strictly divided, each following their own methods and protocols, but concerning themselves with similar issues and topics. Recently, however, composite protocols stemming from the intersection between art and science have been emerging. These composite protocols are relevant to both spheres, but deal with issues unsolvable using methodologies of either sphere separately. It is crucial to search for new knowledge that has references in basic, natural, and applied sciences as well as in art and humanities. To achieve this, we must overcome the persistent modes and patterns of the dualist thinking inherited from Cartesianism as well as abandon the traditional conception that art deals primarily with the aesthetic and beautiful, and that it produces nice, contemplative forms that are made to please our eyes and soul. Such views on art derive from a certain age, i.e. from the 18th and the 19th century when such conception of art flourished, and when the divide between art and science has reached its peak as well. Today, it is time to embark towards a new paradigm of knowledge.

The constructivistic approach we would like to employ here is based on inter-subjectivity instead of the classical objectivity, and on viability instead of reaching one objective truth. It implies that the combination of both scientific fact and artistic/cultural manifestation leads to an abstraction, which can be projected into our cognitive reality. This abstraction of art and science in action is called the composite

projection. The composite projection works as an iteration of the process of extrapolating what we know of reality to what we think reality *should* be, then reconsidering the initial projection with new facts and developments, leading to a modified projection etc. The result therefore has multiple sequential manifestations within the realm of the possible, probable, speculative and fictional. Composite protocols thus stem from both artistic and scientific methodologies, but they are not necessarily consistent with one or the other. They facilitate a holistic understanding of particular topics that are the subject of both science and art practices. The knowledge is generated within the actual/real and conceptual/belief.

In the context of the empirically positivistic conception of science, which operates with the empirically proven, deductive truths, the application of these is guided by necessity, utility and efficiency. The result of such knowledge is therefore an applied solution within the bounds of the possible and measurable. The context of science prohibits the suspension of the possible to construct the impossible, i.e. to produce speculative narratives, fiction and fantasy (as Francis Bacon condemned the philosophy of the speculative as a harmful detour away from the truth). In this sense, speculation (when not understood as extrapolation) and fiction can be conceived as a conscious denial of fact and the reasonable, a state of belief in an idea not embedded in reality, or as a product of the anti-rational. Even so, the futuristic narratives should rely on a consensus of the possible.

Contemporary philosopher Eugene Thacker observes that there have only ever been three approaches to thinking about life: SOUL, MEAT, and PATTERN (Thacker, 2005). Within this trinity everything is deemed to be animate, living, and vital. In the time of networks, swarms, and multitudes of genetic and information technologies, the PATTERN pervades systems of all kinds and it seems to be dominant today. Despite this observation, can we rid ourselves of this trinity and dare to invent some other approach to thinking about life? The existence of our progenies beyond the edges of our heliosphere, in the absolute absence of the human and his/her effects, certainly seeks to broaden the scope of these concepts. What life is in this realm might not fit into Thacker’s trinity of soul, meat and pattern.

### Post-terrestrial Life

From its very beginning, technology has proceeded with the promise of providing us with greater control. Modernity (Modern age) promised control over nature through science



and material abundance through technology. At this point we can find the opportunity to think beyond the confines of control and surveillance, beyond the dualities of utilitarian and non-utilitarian, cause and effect, soul and meat, pattern and random, live and dead, etc. The feedback we can get from technological products that abandon the dualities of our terra-thinking, which in fact owes a lot to Cartesian conceptions, is entirely unpredictable.

The “end-of-life” space objects are terrestrial materializations of human thought having potential to become the emancipated, functional units, capable of cognition and, consequently, of identity. Terrestrial sensory probes at the edges of our solar system, the farthest-reaching manifestations of humanity, were designed to fulfill strictly scientific purposes. However, the ultimate fate of these objects, beyond relentlessly serving humanity with data, had not been determined at their launch. The remoteness and the ebbing life of these extensions of the human species are gradually turning the augmentations into independent objects.

Our challenge is to nurture the teleology of space probes beyond their initial purpose. We aim to explore possible modifications of existing and future space probes to turn allopoietic instruments into resilient, self-repairing, robust, autonomous, energy efficient, adaptable systems, all of which are properties current technology lacks, but living systems possess.

### **The Authentic Environment**

To create artificial life on Earth is a proof-of-principle; proof that we understand living systems to the extent we are able to recreate them. This positivistic approach, however, does answer the question *why* one should attempt to do so, since life has been ubiquitous and resourceful through both space and time in terrestrial environments. Outer space is in fact the authentic environment of artificial life, because life (as far as we know) has not been able to colonize it on its own.

We begin to apply the current knowledge of resilient self-organizing systems to the construction of the next generation of space probes. Unlike scientists, who would equip the next probes with ever better and more complex systems designed to carry out scientific experiments, we foresee the addition of a simple entity, which can withstand the conditions in outer space, but also with the ability to adapt, if it encounters environmental changes; made up of a self-repairable matrix, coined from autonomous technology and simple living (or life-like) systems; working in a symbiosis to absorb entropy and fight decay.

### **Emancipated Space Technologies**

Distance is a tool of scientific and artistic contemplation. Creating progeny that is foreign and non-utilitarian in every respect has great philosophical value as it presents us with a (bio)technological version of the “overview effect”. The alienation induced by this other has the potential to transform the familiar and recalibrate the human condition, urging us to revise the dominant but often exclusionary humanist values.

Humankind, in awe of scientific knowledge, humanistic understanding and artistic possibilities, can produce an

unthinkable civilizational value. Enabling autonomous processes out of our reach is a civilizational step that can lead to a better understanding not only of what we know but also what we don’t.

### **References**

Thacker, E. (2005). Biophilosophy for the 21st Century, 1000 Days of Theory. <http://www.ctheory.net/articles.aspx?id=472>, accessed 10.5.2013

# Evolution of Altruism and Spatial Dispersion: an Artificial Evolutionary Ecology Approach

Jean-Marc Montanier<sup>1</sup>, Nicolas Bredeche<sup>2,3</sup>

<sup>1</sup>NTNU, Trondheim, Norway  
jean-marc.montanier@idi.ntnu.no

<sup>2</sup>UPMC Univ Paris 06, UMR 7222, ISIR, F-75005, Paris, France – <sup>3</sup>CNRS, UMR 7222, ISIR, F-75005, Paris, France  
nicolas.bredeche@isir.upmc.fr

## Abstract

This paper concerns the evolution of altruism in a population of autonomous agents. It explores the relation between altruistic behaviours and spatial dispersion in open-ended evolution whenever energetic constraints must be addressed. A method derived from Embodied Evolution is used to model the spatial interactions between agents from an individual perspective. Firstly, results show that spatial dispersion and levels of altruism are strongly correlated, which confirms theoretical results from biology, but also that this relation may be overshadowed by the complex interactions at work in the ecosystem. Secondly, this paper investigates how robust altruistic behaviours able to cope with various environmental pressures may be evolved. In particular, it is shown that there is a trade-off between efficiency and versatility: the ability to perform well across a wide range of environmental conditions often comes at the cost of sub-optimal performance in terms of survival, especially when compared to more constrained (and less versatile) evolved strategies.

## Introduction

Cooperative behaviours are defined by the realisation of actions by an agent which brings benefits to other agents. This paper focuses on altruism, which is a special type of cooperative behaviour characterised by the sacrifice of one agent for the benefit of others (Lehmann and Keller, 2006). This is different from mutual cooperation as the fitness of an agent is permanently impaired by such acts. This type of behaviour may look counter-intuitive from the viewpoint of the theory of evolution as an altruistic agent reduces its own chance of survival in order to increase the chance of survival of other agents. However, such altruistic behaviour is found in multiple biological species, and multiple works have investigated the conditions and mechanisms at play in its evolution.

A common way to explain the evolution of altruism in biology is to consider the survival of genes rather than individuals. In this context, individuals are vehicles for genes that try to survive (Dawkins, 1976). This aspect is captured by the idea of the inclusive fitness which considers that the fitness of a particular individual depends both from its own action and from the actions of its related kin (Maynard Smith,

1964).

Classic approaches in game theory (Maynard Smith, 1974) and adaptive dynamics (Diekmann, 2004) have been used to explore multiple causes that favour the evolution of altruism. The most studied mechanisms are kin selection (Maynard Smith, 1964), group selection (Wynne-Edwards, 1986), tag recognition (Holland, 1996) and environment viscosity (Hamilton, 1964). In particular, kin selection stresses that genes responsible for altruistic behaviours can increase in frequency when there is a chance that beneficiaries of such altruistic acts also carry such genes. In other words, kin selection hypothesizes that the inclusive fitness of an individual is increased if it is genetically close to its neighbours. Kin selection has long been a central idea in the evolution of altruism, and recent works have shown that several other mechanisms (such as group selection) are actually much more related to kin selection than originally expected (West et al., 2007; Grafen, 1984; Queller, 1994; van Baalen and Rand, 1998). Moreover, explicit behavioral strategies have been shown to increase kin selection, such as kin recognition and spatial dispersion (West et al., 2007). In particular, a low spatial dispersion naturally favors reproduction among kins.

From the perspective of artificial evolution, several works have previously addressed the evolution of altruism (Waibel et al., 2009), and of communication (Floreano et al., 2007) (a particular kind of cooperative behaviour) with regards to the level of selection (at the level of the team or the individual), and the composition of teams (homogeneous or heterogeneous). These works successfully show cooperative behaviours could evolve from team level selection or by enforcing homogeneous teams, which is coherent with results previously established in theoretical biology (cf. (Hamilton, 1964)). However, these works rely on a fixed selection scheme (rather than letting it evolve) which prevents studies of particular dispersion strategies that could influence the level of homogeneity and relatedness in the population.

This paper addresses the evolution of spatial dispersion behaviour, in the context of a harvesting task that requires altruistic cooperation among individuals. The question under

scrutiny is to understand how spatial dispersion may evolve when altruistic behaviour comes as a requirement for the population to survive. In particular, it is expected that there is a correlation between the consumption strategy and the spatial dispersion evolved: the more altruistic the individuals, the less spatial dispersion should be observed (Taylor, 1992)). However, questions remain open as to whether such behaviours may be observed easily in nature, and what kind of behaviours may be evolved in term of consumption and spatial dispersion strategies.

The approach followed in this work builds on an existing framework for *in silico* experimental evolution for individual-based modeling and simulation undergoing an open-ended evolutionary process (i.e. long term adaptation in an open environment). In this context, the ability for an individual to survive and pass its genotypic material depends solely on its interaction with other individuals and with the environment, comparably to Dawkins' selfish gene metaphor (Dawkins, 1976) or TIERRA's open-ended evolutionary process (Ray, 1992). Therefore, it is possible to investigate the particular dispersion strategies which comes from a trade-off between harvesting and genotypic material diffusion.

In the following, the experimental setup is described, along with methodological tools and implementation details. A statement of the working hypotheses and outline of the experiments follows. Then, the experiments are described and discussed. Firstly, the possible correlation between spatial dispersion and level of altruism is investigated. Secondly, the trade-off between evolving either efficient or versatile strategies is studied. Finally, the last Section concludes this work and takes a broader perspective from this work, considering implications both from the theoretical viewpoint wrt. biology and from the practical viewpoint wrt. collective adaptive systems.

## Method

### Open-ended Evolution with mEDEA

The **mEDEA** algorithm, as in *minimal* Environment-driven Distributed Evolutionary Adaptation, was initially introduced in (Bredeche and Montanier, 2010). It performs as an evolutionary adaptation algorithm that can be distributed over a population of agents (i.e. each agent in the population runs the same algorithm, but carries different genomes). While it has been originally designed for collective robotic systems, it can be (and has been) used as a modeling and simulation tool for studying spatial interactions between agents. In previous works (cf. Montanier and Bredeche (2011)), mEDEA has been used to study the impact of genotypic relatedness on altruistic cooperation, in particular whenever genotypic relatedness between individuals is enforced through kin recognition (i.e. explicitly favoring the reproduction of closely related individuals).

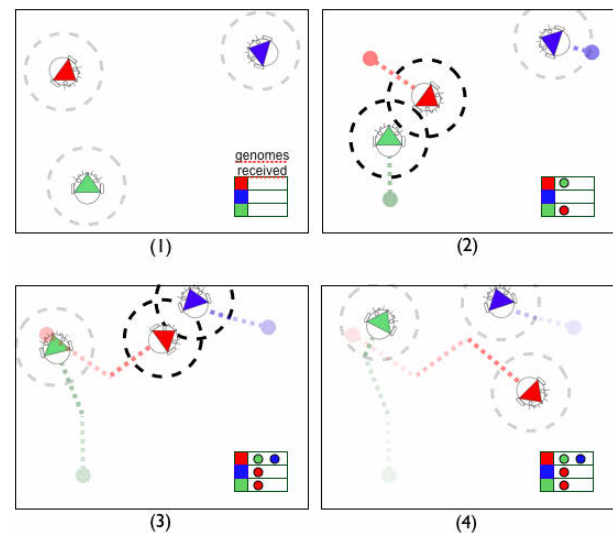


Figure 1: The mEDEA algorithm: a simplified illustration. (1): generation starts, genome reservoirs are empty; (2) and (3): agents move around (each agent is controlled by its own active genome) and exchange mutated genomes when close enough; (4): generation ends - the red genome has spread more and thus have higher probability of being selected (in this case, probability is indeed  $p = 1$  in two agents while the two other genomes only get  $p = 0.5$  in one single agent). After selection, all the reservoirs are emptied. Note that the next generation will contain slightly mutated copies of the original genomes.

Figure 1 provides an illustrative example of how mEDEA works (see (Bredeche and Montanier, 2010) for a complete description of the algorithm). Each robotic agent contains an *active* genome, which (indirectly) controls the agent's behaviour, and a *reservoir of stored genomes*, which is empty at first. At each time step (or iterations), each agent *broadcasts* in a limited range a *slightly* mutated copy of its active genome (gaussian mutation) and stores genomes received from neighbours, if not already stored. At the end of a generation (i.e. a pre-defined number of iterations), each agent "forgets" its active genome and *randomly* picks one genome from its reservoir of stored genomes (if not empty). Then the reservoir is emptied, and a new generation starts. This algorithm is running independently within each agent in the population. By this mean, agents' behaviours differ depending on each agent's current active genome.

Therefore, selection pressure occurs at the population level (the more a genome spreads itself, the higher the probability it will generate offsprings) rather than at the individual level (random sampling). Genomes survive only through spreading (as an active genome is automatically deleted locally at the end of a generation) and individual may get better over time as conservative mutations generate new candidates that explore alternative behavioural strategies.

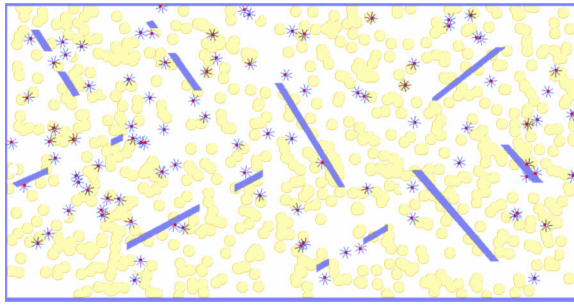


Figure 2: Snapshot from the simulator: food items (circles), agents (dots) and obstacles

On the one hand, mEDEA resembles other open-ended evolutionary setups such as TIERRA (Ray (1991)) as selection pressure occurs through interactions between agents in their environment rather than by explicitly computing a fitness value (such as a metabolic function). On the other hand, relying on a maximum number of active mobile agents makes it possible to set an upper bound in term of computational time required for the simulation, in a similar fashion as it is by setting a maximum number of cells in AVIDA (Ofria and Wilke, 2004). In mEDEA, each genome competes for accessing a limited set of resources: the population of robotic agents.

### Setup

The setup used in this work (displayed in Figure 2) features simulated robotic agents which harvest food items from the environment in order to remain active. Each robotic agent consumes a fixed amount of energy at each iteration, and has a limited energy storage capability.

Each agent can be in three different states: active, dead or listening. At the beginning of a run all agents are in the active state, i.e. they are using an active genome (randomly generated in the range  $[-1.0; 1.0]$ ), and their energy level is greater than 0. If during a generation an agent runs out of energy, it switches to the dead state. In this state the agent has no active genome, remain stationary, and cannot store genomes from robots passing by. The dead state is maintained for one generation, after which the agent switch to a listening state. In this state the agent doesn't move but stores the genomes broadcasted by agents in its neighbourhood. This state is maintained during one generation. If at the end of a generation in listening state the reservoir is still empty, the agent will remain in listening state for another generation, and so on. Any agent with an empty reservoir at the end of a generation switches to the listen state.

Food items are randomly placed in the environment. Once a food item has been harvested it becomes unavailable for some time, termed  $EP_{Lag}$  (the regrow delay). This term depends linearly on the energy harvested from it as shown in Equation 1:

$$EP_{Lag} = E_{harvested} / EP_{eMax} * EP_{LagMax} \quad (1)$$

$EP_{eMax}$  is the maximal amount of energy that can be harvested from a food item by an agent.  $E_{harvested}$  is the energy actually harvested from the food item by an agent.  $EP_{LagMax}$  is the maximal regrow delay of a food item.

Environmental pressure can be changed from low to high by setting the value of the  $EP_{LagMax}$  parameter. Large  $EP_{LagMax}$  values result in longer regrow delay (i.e. larger  $EP_{Lag}$  values) whenever a food item is completely harvested, which decreases the number of food items available for some time.

### Monitoring Consumption Strategy

An agent may display an altruistic behaviour by harvesting only part of a food item. Such a consumption strategy is costly in terms of fitness (as it might run out of energy), and is of benefit to other agents (the food item will regrow faster). On the contrary, selfish agents will completely harvest any food item, which is likely to increase their chance to survive, but also reduces the number of food items available to other agents.

The consumption cost an agent accepts to pay is measured by the difference between how much energy could have been harvested by the agent (in order to completely fill the battery), and how much was really harvested. Equation 2 gives a definition of the consumption cost:

$$Cost = \max(0, \min(EP_{eMax}, r_{E_{max}} - r_{E_{now}}) - E_{harvested}) \quad (2)$$

$EP_{eMax}$  is defined as before (i.e. maximal energy in a food item),  $r_{E_{max}}$  is the maximal energy level of an agent,  $r_{E_{now}}$  is the current energy level of the agent, and  $E_{harvested}$  is the energy harvested by the agent from the food item.

While a selfish agent shall have a consumption cost of zero, an altruistic agent should ideally be able to perform a trade-off between its altruistic nature and its survival needs. Therefore, the consumption cost of altruism can be seen as the agent's level of sacrifice which is continuous (a quantity of energy) rather than discrete (eat or don't eat).

As a last remark it should be noted that the consumption strategy is but one way to monitor altruistic behaviours. As an example, two different consumption strategies, each combined with a different exploration strategy (travelling speed, area coverage) may well end up with the same number of food items available at any time (slow but greedy vs. fast but frugal agents). The next paragraph investigates how to take into account spatial dispersion strategies, in addition to consumption strategies already considered.

### Monitoring Spatial Dispersion

Spatial dispersion may impact harvesting strategies as well as altruistic cooperation in various way as low dispersion



(i.e. remaining in the same region) may both favour exploitation of the same food items as well as increasing kin selection (and therefore impact the level of altruism, as long known since Maynard Smith (1964)). In order to account for spatial dispersion, we devise a measure to approximate the area covered by one agent during its lifetime.

In order to do so, the environment is divided into squared regions. For each agent, the number of regions is counted during its lifetime, and the *AreaCovered* value is computed following Equation 3:

$$AreaCovered = \frac{\#VisitedCells}{\#Cells \times lifetime} \quad (3)$$

In the following, the environment has been divided in 33920 cells of 4 by 4 pixels. The theoretical minimal value of this measure is  $1/(\#Cells \times lifetime) = 7.34258 \times 10^{-08}$ . The maximal value depends on agents maximal speed ( $Max_{Agent_{speed}}$  given in fraction of cell area), and is given by Equation 4:

$$Max_{AreaCovered} = \frac{Max_{Agent_{speed}}}{\#Cells} \quad (4)$$

### Working Hypotheses

Firstly, the link between consumption strategies and spatial dispersion will be investigated by monitoring spatial dispersion whenever a pre-defined consumption strategy is used. The expected result is that the level of altruism displayed during consumption of food items should be (negatively) correlated with spatial dispersion (the higher the consumption cost, the lower the dispersion) as lower dispersion theoretically increases genotypic relatedness, which is a key to altruistic behaviour. However, it remains to be investigated if such results can actually be observed as survival becomes more challenging as environmental pressure increases (i.e. spatial dispersion may not be solely driven by altruistic motivation).

In practical, this will be done by enforcing the amount of energy that is *left* when harvesting a food item. In Equation 2, this corresponds to setting a value for  $E_{harvested}$  so that the *Cost* paid is equal to the "fixed" cost expected. In this paper, two different fixed costs, each close to one particular extreme consumption behaviour, are investigated: whenever a food item is harvested, either 5 (slightly altruist) or 40 (very altruist) units of energy are left over on a total of the 50 units of energy a food item can provide, implying different consequences on the food item's delay to regrow. These *fixed cost* consumption strategies will be referred to as *cost* = 5 and *cost* = 40 consumption strategies in the next Section.

Secondly, the possible benefits of leaving to evolution both the consumption strategy *and* the spatial dispersion strategy will be investigated. It is indeed not clear that letting

both strategies evolve should lead to better survival strategies, as evolution may face a more difficult challenge due to an increased number of degrees of freedom. In practical, the consumption cost to be paid when a food item is harvested will be left to the robot to decide and both consumption cost (cf. Equation 2) and spatial dispersion will be monitored. The expected result is that evolving both consumption cost and spatial dispersion may possibly lead to a richer set of behaviours whenever environmental pressure varies, though possible benefits remain to be identified. This consumption strategy will be referred to as *dynamic cost* in the next Section.

## Results

### Technical Details

A Multi-Layer Perceptron (MLP) is used to encode the controller of each robotic agent. The input layer is composed of 12 inputs (8 for distance sensors, 1 for the direction to the closest energy point, 1 for the distance to the closest energy point, 1 for the battery level of the agent, 1 to detect the presence of an energy point under the agent), the hidden layer is composed of 5 neurons, and the output layer is composed of 3 neurons (rotational speed, translational speed and amount of energy to be harvested (used only if a food item is within reach)). The output neuron for energy harvesting is not taken into consideration when a fixed cost is used. The weights of the MLP are decoded from the active genome of the agent. A gaussian mutation is used, and initial weights are set randomly around zero. The  $\sigma$  parameter for mutation is evolved, and a minimal value (0.01) is fixed to avoid obtaining a population of clones

All experiments are performed with RoboroBo, a fast open-source multi-robot simulator (Bredeche et al., 2013). In order to ensure the reproducibility of the experiments presented in this paper, the full implementation is available online<sup>1</sup> and parameters used are summarized in Table 1. One run takes approximately one hour to be performed using one core of a quad-core 2 CPUs Intel 2.26 GHz processor. All experiments presented in this paper are performed on a computer cluster equipped with such processors. For each setup considered in the next section results, each Figure results from a compilation of 500 independant runs, and statistical significance is tested using the Wilcoxon signed-rank test (Wilcoxon, 1945).

### Evolution of Spatial Dispersion Strategies

In order to obtain results on a large range of environmental pressures, the experiment starts with a low environmental pressure ( $EP_{Lag_{Max}} = 25$  iterations) until the 400000<sup>th</sup> iteration. After this, the environmental pressure

<sup>1</sup>[http://pages.isir.upmc.fr/evorob\\_db/moin.wsgi](http://pages.isir.upmc.fr/evorob_db/moin.wsgi)

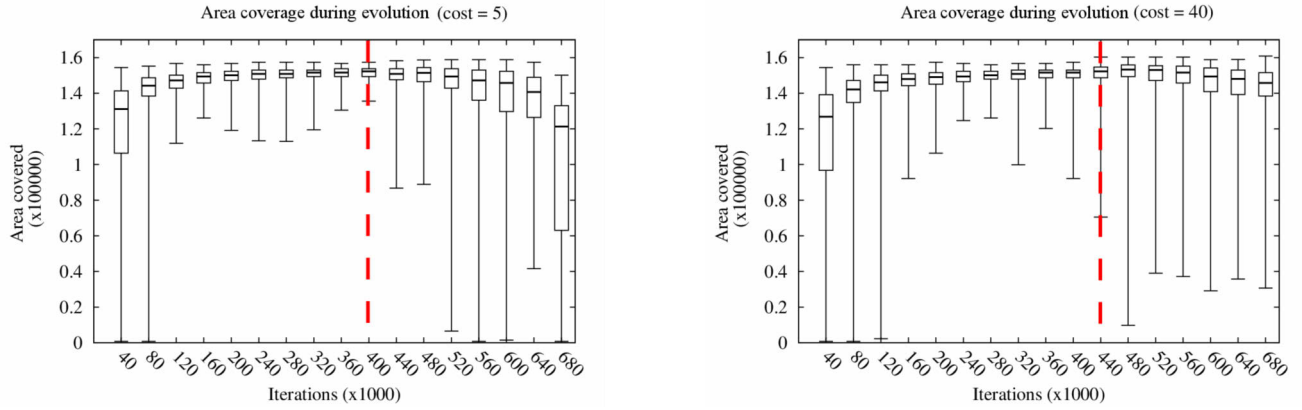


Figure 3: Area coverage measured when the cost of altruism is fixed to 5 (left) and 40 (right) and the pressure of the environment increases by step of 80 iterations every 4000 iterations from iteration 400000 (before the environmental pressure is  $EP_{LagMax} = 25$  iterations).

Parameter	Value
arena width and length	$1024 * 530$ pixels
lifetime (i.e. generation duration)	400 iterations
selection scheme	random
population size	100 agents
agent size	1 pixel
proximity sensor range	64 pixels
radio broadcast signal	32 pixels
agent rotational velocity	0.52 rad/iteration
agent translational velocity	3 pixels/iteration
genome length	84 real values (83 MLP weights + $\sigma$ )
variation operator	Gaussian mutation with $\sigma$ parameter
cells size	4 by 4 pixels
theoretical max area	$2.21108 \times 10^{-5}$
theoretical min area	$7.34258 \times 10^{-8}$
$EP_{LagMax}$	50
number of energy points	800
$r_{eMax}$	400
energy consumption	1 per iteration

Table 1: Parameters for experiments.

slowly increases every 4000 iterations (10 theoretical generations) until the population goes extinct (i.e. no genome left to exchange). Each increase of the environmental pressure is done by a fix amount of 80 iterations in the regrow delay ( $EP_{LagMax}$ ). As an example,  $EP_{LagMax} = 105$  iterations at the 404000<sup>th</sup> iteration of the simulation, and  $EP_{LagMax} = 185$  iterations at the 408000<sup>th</sup> iteration.

Results obtained when two harvesting strategies with fixed costs of 5 (less altruistic strategy) and 40 (more altruistic strategy) are used are presented in Figure 3 (500 runs for each setup). With both strategies the area dispersion evolved is increasing until iteration 400000<sup>th</sup> and decreasing after ( $p$ -value  $< 0.05$  for comparison of iteration 400000 and every iteration after 560000). This shows that different spatial dispersion strategies are displayed through evolution depending on the consumption strategy used and the environmental pressure at hand.

The differences between spatial dispersions evolved under different consumption strategies is expected from results obtained in biology (as said before). However, results shown here are contradictory with theory: spatial dispersion is shown to be higher for the more altruistic con-

sumption strategies when challenging environment are considered (while kin selection, favored by lower dispersion, should be paired with an increased altruistic behaviour (Taylor, 1992)). Rather than contradicting well established theoretical results, individual based modeling and simulation actually points out the complex interactions between individuals and the environment. Indeed, dispersion strategies may be influenced by much more than just acting on genotypic relatedness. The number of active agents, the availability of energy points, and the regrow delay are all possible causes to explain a particular dispersion strategies. Therefore, one question remains: in a comparable setup (i.e. removing all other possible causes), how does dispersion strategies compare when evolved with different fixed consumption strategies.

### Fair Comparison of Dispersion Strategies

In order to compare results from the two setups considered previously, agents' spatial dispersions are measured in a similar environment. The environment used for comparison features a consumption cost artificially fixed to 0 (whatever the initial consumption cost used during evolution) and a low environmental pressure ( $EP_{LagMax} = 25$  iterations). Moreover, genome transmission and selection are shut down, and robots continue to run even if energy is depleted. This makes it possible to compare the different behavioural strategies by *replaying* evolved genomes with all other parameters set to similar values. The following replay procedure is defined: (1) genomes from the 600000<sup>th</sup> iteration of a given run are randomly sampled to assemble a population of 100 individuals ; (2) this population is embodied in 100 robots (one genome per robot) (3) The spatial dispersion of these robots is measured during 40000 iterations.

For each fixed cost strategy considered earlier, genomes

are extracted from 20 runs selected randomly among the 500 discussed earlier, and replayed following the procedure described above. The replay procedure is repeated 20 times for each setup (total of 800 replays). In addition, we devise a third type of behaviour, which stands as a control experiment: agents using random controllers. Random translational and rotational speeds are assigned to 100 agents at each iteration, and spatial dispersion is measured for 40000 iterations. This is inspired from typical tests in ecology field studies where geographic tracking of animals positions are evaluated wrt. brownian motion (Borger and Fryxell, 2012). This control experiment is termed *random behaviour*, and is also performed 800 times.

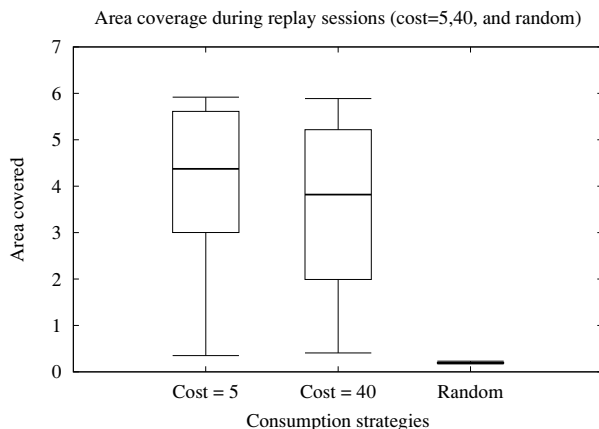


Figure 4: Area coverage for three consumption strategies when replayed, and for runs using random movements. All area coverage are significantly different:  $p - value < 0.05$  for comparisons between each boxplot.

The median results for each fixed consumption strategy and the random behaviour are presented in Figure 4. As expected, both strategies evolved under environmental pressure (cost fixed to 5 and cost fixed to 40), display higher level of dispersion than random movements. Then, comparing the fixed cost consumption strategies, it is shown that the area covered is significantly higher when a lower level of altruism is used during evolution ( $p - value < 0.05$  between a cost fixed to 5 and a cost fixed to 40). As a consequence, this confirms expected results from theory: there is indeed a negative correlation between spatial dispersion and consumption behavior: enforcing a consumption strategy which displays altruistic behaviour leads to lower spatial dispersion, which is expected to increase kin selection, and thus altruistic behaviour among closely related individuals.

Typical runs for each behaviour studied (i.e. runs at the median), are shown in Figures 5(a), and 5(b). In these two Figures, which can only be interpreted in light of the previously shown quantitative analysis, illustrate slight but visible differences in spatial dispersion. Trajectories obtained with a highly altruistic consumption strategy ( $Cost = 40$ ) dis-

play more localized behaviours (i.e. robot circling around in the same area) and fewer wandering trajectories. Counting the average number of encounters per agent also advocates for local interactions: there are significantly less encounters for the  $Cost = 40$  strategy than for the less altruistic, more exploratory,  $Cost = 5$  strategy ( $p - value < 0.05$ , Wilcoxon test).

### Evolving Consumption and Dispersion Strategies

In this last part of the paper, we investigate the impact of evolving both the consumption strategy and the dispersion strategy. By doing so, we intend to address the following questions: (1) What kind of (consumption and dispersion) strategies can be expected when evolved under different environmental pressures ; (2) What are the possible benefits and drawbacks of evolving the consumption strategy rather than enforcing an ad hoc consumption strategy.

As before, 500 runs are performed. The setup is similar to the previous setups for fixed cost, except that the cost of altruism is now chosen by the robot controller. This setup is termed "Dynamic Cost" has the cost paid may change any-time and depends from evolution (i.e. the controller output fixing the amount of energy taken from a food item is actually used). Figure 6 shows the boxplot results for consumption costs paid and area dispersion evolved by all agents throughout evolution. As before, the environment becomes gradually more challenging starting iteration 400000, and stops when all runs have gone extinct.

A notable difference is that during the first part of the runs, the consumption costs paid stick to zero, which is not unexpected as there is no benefit at being altruistic in an environment that represent an easy challenge. The consumption cost paid then abruptly changes as soon as the environmental pressure increases ( $p - value < 0.05$  for comparison between iteration 400000 and iteration 440000). It then fluctuates around a value of 5 until iteration 680000, and remains significantly higher than at the beginning of the run ( $p - value < 0.05$ , comparing results from iteration 400000 and any iteration afterwards). Moreover, the final value (at iteration 680000) is similar to the value obtained at iteration 440000 ( $p - value = 0.31$ ). Hence, there appears to be two stable values (either no altruism ( $Cost = 0$ ) or low altruism ( $Cost \approx 5$ )) for consumption cost depending on the challenge posed by the environment.

Regarding spatial dispersion, Figure 6-right shows that the area covered by each agent levels up, and then off, until iteration 400000. Then, as environmental pressure starts to increase, the area covered is continuously decreasing, and ends up as significantly lower beyond iteration 560000 ( $p - value < 0.05$ , comparing area covered at iteration 400000 and any iteration from 560000).

In order to compare the behaviours obtained with a dynamic cost, replay sessions are performed in the exact same fashion as it was for the fixed cost setups. Quantitative re-

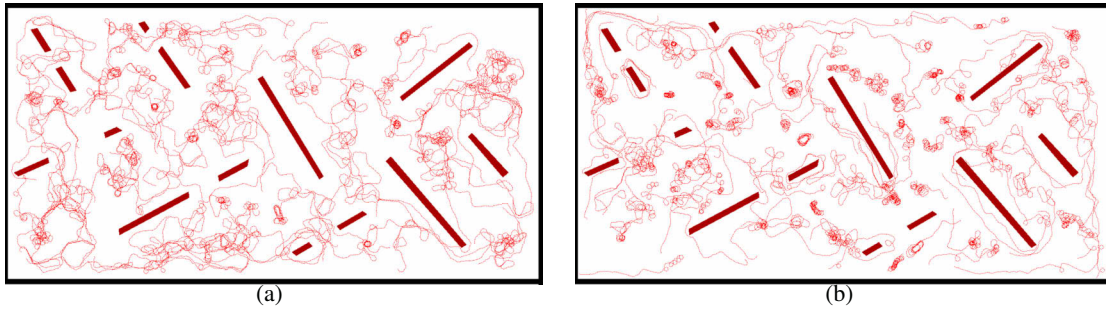


Figure 5: Trajectories from median runs for consumption strategies of cost=5 (left) and cost=40 (right).

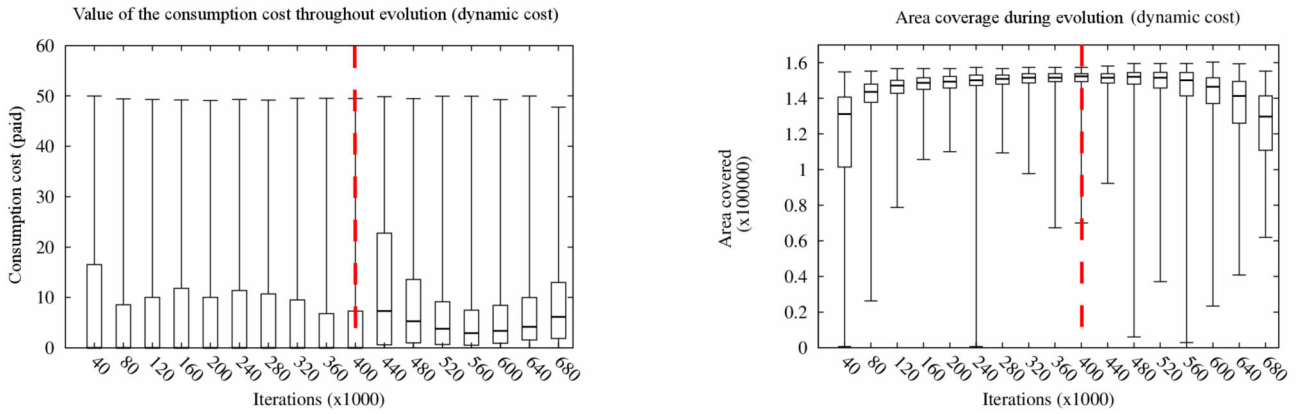


Figure 6: Consumption cost of altruism (left) and area coverage (right) measured when the Consumption cost is dynamic and the pressure of the environment increases by step of 80 iterations every 4000 iterations from iteration 400000 (before the environmental pressure is  $EP_{LagMax} = 25$  iterations).

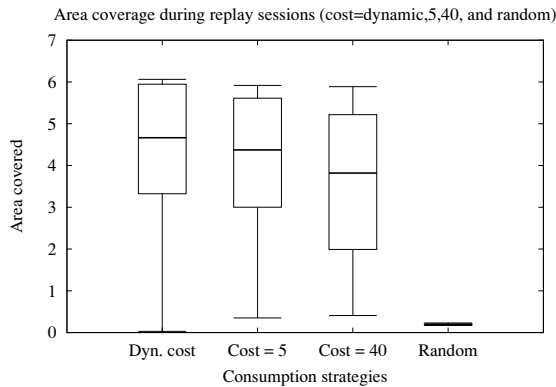


Figure 7: Area coverage for two consumption strategies when replayed, and for runs using random movements. All area coverages are significantly different.

sults obtained by all consumption strategies studied are presented in Figure 7. Results obtained with the dynamic cost strategy is different from all other strategies ( $p$  - value < 0.05 for comparisons between each cost strategy). This con-

firms the impact of cost strategies on the evolution of behaviours.

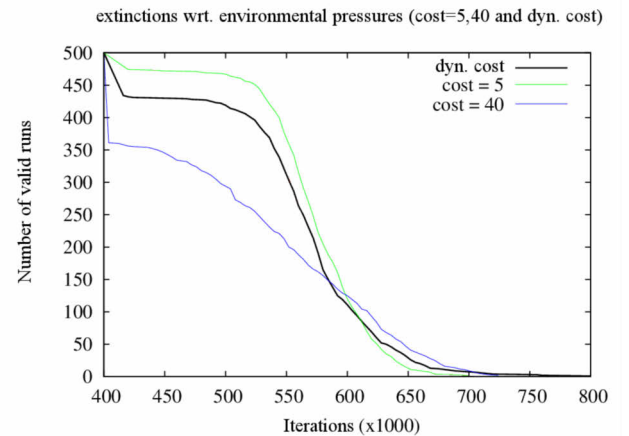


Figure 8: Number of active runs when the environmental pressure is increasing.

Another way to study the differences between different cost strategies is to observe the extinction of runs : that is,



for a given iteration, how many runs is there with at least one active agent. Figure 8 shows the number of active runs for all three setups considered ( $Cost = 5$ ,  $Cost = 40$  and dynamic Cost) starting iteration 400000, that is when environmental pressure starts to gradually increase. From this figure, several particular situations arise: the  $Cost = 5$  strategy dominates before iteration 580000, and is replaced by the  $Cost = 40$  strategy around iteration 600000 and until iteration 700000. Afterwards, only the  $Cost = 40$  and dynamic Cost remain for another 100000 iterations until all runs for both go completely extinct. Though the dynamic Cost strategy goes a little further than the  $Cost = 40$  strategy, there are too few runs left to make any statistically significant remark between the two setups beyond iteration 700000. Most notably, the dynamic Cost strategy is nearly never completely dominated by the two other strategies. As such, it appears that relying on a dynamic cost strategy provides some kind of trade-off between optimal performance in specific contexts and versatility in various contexts, without having to take the risk of guessing which cost must be paid prior to evolution.

## Conclusion

In this paper, we have studied the importance of spatial dispersion during the evolution of altruistic cooperation in a population of autonomous agents. This work followed an individual-based modelling and simulation approach using the mEDEA algorithmic framework for open-ended evolution, which enables to study the actual evolution of individual spatial dispersion and consumption strategies in the context of a harvesting task.

Firstly, well-established theoretical results on kin selection in the evolution of altruism were confirmed experimentally regarding the negative correlation between the level of altruistic behaviour and spatial dispersion. However, our work revealed that such a confirmation may not come for free when individual modeling and simulation is considered as the complex interactions between the population and its environment may provide contradictory results at first glance. This result is important as it may have an impact on field observation from nature, where the level of spatial dispersion may fail to directly explain the occurrence of altruistic behaviour.

Secondly, we showed that there is a trade-off between setting ad hoc mechanisms (here, the strategy used when eating a food item) and letting such mechanisms to evolution. On the one hand, results show that the more there is left to evolution, the less likely optimal behaviour may be reached (compared to *a priori* carefully crafted strategies). On the other hand, fully evolved strategies turn out to be more versatile, i.e. showing good performance in a larger set of contexts, and require less prior knowledge compared to more constrained evolutionary setups.

Lastly, this paper intends to contribute both to theoretical biology, by providing new results from an individual-based

modeling perspective where spatial dispersion is the product of open-ended evolution, and to collective adaptive systems, as the algorithm used throughout this paper may straightforwardly be implemented onto real robots (and has already been, albeit for a different problem).

## Acknowledgments

Experiments presented in this paper were carried out using the Grid'5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see <https://www.grid5000.fr>). This work was made possible by the Alain Bensoussan Fellowship Programme.

## References

- Borger, L. and Fryxell, J. (2012). *Quantifying individual differences in dispersal using the net squared displacement*, chapter 17. Oxford University Press, Oxford (UK).
- Bredecche, N. and Montanier, J.-M. (2010). Environment-driven Embodied Evolution in a Population of Autonomous Agents. In *The 11th International Conference on Parallel Problem Solving From Nature (PPSN 2010)*, pages 290–299.
- Bredecche, N., Montanier, J.-M., Weel, B., and Haasdijk, E. (2013). Roborobo! a fast robot simulator for swarm and collective robotics. *CoRR*, abs/1304.2888.
- Dawkins, R. (1976). *The Selfish Gene*, volume 32. Oxford University Press.
- Diekmann, O. (2004). A beginner's guide to adaptive dynamics. *Mathematical Modelling of Population Dynamics*, 63:47–86.
- Floreano, D., Mitri, S., Magnenat, S., and Keller, L. (2007). Evolutionary conditions for the emergence of communication in robots. *Current Biology*, 17(6):514–519.
- Grafen, A. (1984). Natural selection, kin selection and group selection. *Behavioural ecology: an evolutionary approach*, 2nd edition, pages 62–84.
- Hamilton, W. (1964). The genetical evolution of social behaviour. *Journal of Theoretical Biology*, 7(1):1–16.
- Holland, J. (1996). *Hidden Order: How Adaptation Builds Complexity*. Basic Books.
- Lehmann, L. and Keller, L. (2006). The evolution of cooperation and altruism – a general framework and a classification of models. *Journal of Evolutionary Biology*, 19(5):1365–1376.
- Maynard Smith, J. (1964). Group selection and kin selection. *Nature*, 201:1145–1147.
- Maynard Smith, J. (1974). The theory of games and the evolution of animal conflicts. *Journal of theoretical biology*, 47(1):209–221.
- Montanier, J.-M. and Bredecche, N. (2011). Surviving the tragedy of commons: Emergence of altruism in a population of evolving autonomous agents. In *Proceedings of the 11th European Conference on Artificial Life (ECAL'11)*, pages 550–557.
- Ofria, C. and Wilke, C. O. (2004). Avida: A software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229.
- Queller, D. (1994). Genetic relatedness in viscous populations. *Evolutionary Ecology*, 8(1):70–73.
- Ray, T. S. (1991). An approach to the synthesis of life. In Langton, C., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, volume XI of *Santa Fe Institute. Studies in the Sciences of Complexity*, page 371408. Addison-Wesley, Redwood City, CA.
- Ray, T. S. (1992). Evolution, ecology and optimization of digital organisms. Technical report, Santa Fe Institute.
- Taylor, P. D. (1992). Inclusive fitness in a homogeneous environment. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 249(1326):299–302.
- van Baalen, M. and Rand, D. (1998). The unit of selection in viscous populations and the evolution of altruism. *Journal of theoretical biology*, 193(4):631–648.
- Waibel, M., Keller, L., and Floreano, D. (2009). Genetic team composition and level of selection in the evolution of cooperation. *IEEE Transactions on Evolutionary Computation*, 13(3):648–660.
- West, S. A., Griffin, A. S., and Gardner, A. (2007). Social semantics: altruism, cooperation, mutualism, strong reciprocity and group selection. *Journal of Evolutionary Biology*, 20:415–432.
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.
- Wynne-Edwards, V. C. (1986). *Evolution through group selection*. Blackwell Scientific, Oxford.

# Environmental Feedback Drives Multiple Behaviors from the Same Neural Circuit

Paul L. Williams<sup>1</sup> and Randall D. Beer<sup>1,2</sup>

<sup>1</sup>Cognitive Science Program and <sup>2</sup>School of Informatics and Computing  
Indiana University, Bloomington, IN 47406  
plw@indiana.edu

## Abstract

The ability of a single neural circuit to produce qualitatively distinct behaviors is typically attributed to some adaptive mechanism in the circuit itself. However, neural circuits are also embedded in particular bodies and environments, and feedback through the sensorimotor loop may also serve to drive behavioral differentiation. Here we explore the ability of a single neural circuit to produce qualitatively different behaviors based on changing patterns of environmental feedback. Agents equipped with two sets of effectors and controlled by fixed neural circuits are evolved to catch circles under three different motor conditions. In one condition, the agent must coordinate both sets of effectors, while in each of the other conditions one set of effectors is lesioned and the agent must rely on the other set alone to accomplish the task. A detailed behavioral analysis of the best evolved agent is reported, providing numerous insights into its evolved behavioral mechanism. The agent is found to produce significantly different motor outputs in each of the three conditions, to rely on continuous environmental feedback for successful behavior, and to switch flexibly between different behavioral conditions.

## Introduction

The ability of a single neural circuit to produce multiple qualitatively distinct behaviors, referred to as *multifunctionality*, is typically thought to be due to some adaptive mechanism in the neural circuit itself. For example, a single neural circuit may produce multiple distinct behaviors as a result of synaptic plasticity, neuromodulation, or intrinsic multistability (Briggman and Kristan, 2008; Getting, 1989; Morton and Chiel, 1994). In all of these mechanisms, the primary source of behavioral differentiation is assumed to be the neural circuit itself, while the role of bodily and environmental context is taken to be of secondary importance. However, in the past few decades, researchers from a variety of disciplines—including artificial intelligence, neuroscience, philosophy of mind, and cognitive science—have increasingly emphasized the importance of *situatedness* and *embodiment* for the production of intelligent behavior (Brooks, 1991; Clark, 1995; Pfeifer and Bongard, 2007; Beer, 2008). Broadly speaking, situatedness refers to the role played by

an agent's ongoing interactions with its immediate environment in shaping behavior. For example, a situated agent may substitute actions in the world for actions in the head, effectively offloading aspects of cognitive processes to the environment (Kirsh and Maglio, 1994; Hutchins, 1995). Embodiment refers to the influence that the structure and properties of an agent's body have on its behavior. For instance, embodiment allows an agent to actively select and structure the information that it receives from its environment (Lungarella and Sporns, 2005; Pfeifer et al., 2007; Polani et al., 2007).

But how much can situatedness and embodiment really influence behavior? In particular, can different bodily or environmental contexts produce qualitatively different behaviors from the same neural circuit, or only slight variations? As a corollary, how important is it for cognitive scientists to take into account the bodies and environments of intelligent agents in order to understand the mechanisms that produce their behavior? A recent study by Izquierdo and Buhrmann (Izquierdo and Buhrmann, 2008) explored these questions in a radical way, by evolving model neural circuits to exhibit qualitatively distinct behaviors when their bodies and environments were literally switched. Specifically, building upon earlier studies where neural circuits were evolved for walking (Beer and Gallagher, 1992; Beer, 1995a; Beer et al., 1999) and chemotaxis (Beer and Gallagher, 1992), Izquierdo and Buhrmann evolved individual circuits to perform both tasks. In one condition, the neural circuits were embodied in a simple legged agent and evolved to exhibit walking behavior. In a second condition, the same neural circuits were embodied in an agent with a chemo-sensor and were evolved to perform chemotaxis. Crucially, the neural circuits were evolved with fixed synaptic weights, so that there was no intrinsic adaptive mechanism in the circuits themselves. Additionally, the circuits did not receive any explicit signal indicating which of the two behavioral conditions they were in. Thus, the only information that the circuits received about the appropriate behavior for their current context, and the only means by which the circuit could generate these distinct behaviors, was via changing patterns of feedback through the body and environment.

Izquierdo and Buhrmann also performed a dynamical analysis of the best evolved circuit, and found that the same region of autonomous dynamics was utilized by the circuit in both behavioral conditions. In other words, the different behaviors of the circuit could not be tied to different regions of autonomous dynamical behavior, as one might conventionally expect. Rather, the distinct behavioral patterns were shown to arise solely through different patterns of feedback on multiple timescales within the behaving brain-body-environment system. Thus, the primary objective of this study was to demonstrate the profound importance that feedback through the body and environment can have on producing different behaviors from the same neural circuit, and in this it succeeds admirably.

However, it would also be desirable to demonstrate the same idea without requiring the radical changes of swapping bodies and environments, especially since biological circuits exhibit multifunctionality often times without such changes. In particular, one would like to demonstrate that changing environmental feedback alone suffices to evoke different behaviors, while body and environment remain the same. In general terms, behavior is the product of a complete brain-body-environment system, and thus the generation of multiple behaviors can be driven by changes in any part of that system. Thus, while multiple behaviors can be produced, as in the case of Izquierdo and Buhrmann, by changing multiple parts of the brain-body-environment system simultaneously, multiple behaviors can also be produced by changing only one element of the brain-body-environment system at a time. In addition to changes in the underlying neural system (Briggman and Kristan, 2008), such changes can also include modifications to an agent's sensors (Buhrmann et al., 2013), actuators (Nolfi, 2009), or overall body morphology (Fine et al., 2007; Auerbach and Bongard, 2009).

The goal of this project was to achieve a simple demonstration of how changes in an agent's actuators can drive multiple behaviors, and to begin exploring exactly how multiple behaviors can be produced through changing patterns of environmental feedback. In this study, agents are evolved to catch circles falling towards them from above. The agents are equipped with two set of effectors, and are evolved to catch circles under three behavioral conditions. In the first condition, the agent must coordinate the actions of both sets of effectors to successfully catch the objects. The other two conditions are formed by "lesioning" one or the other of the agent's effectors, such that the agent must perform the task with only one functional set of effectors. Thus, in each of the three conditions, the agent must generate radically different behaviors in order to accomplish the task. Moreover, as in (Izquierdo and Buhrmann, 2008), the neural circuit controlling each agent has fixed synaptic weights, and the agent receives no direct information regarding which or the three conditions it is in, i.e., there is no signal indicating that one or the other of its effectors have been lesioned. As a result,

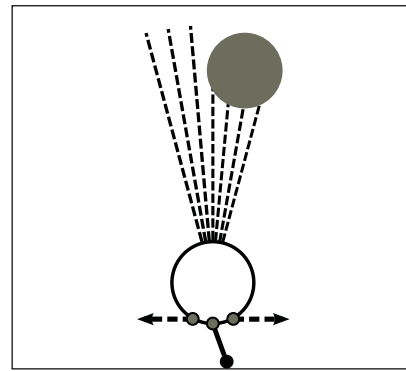


Figure 1: The agent and environment. The agent moves horizontally using two sets of effectors while circles fall towards it from above. The agent's sensory apparatus consists of an array of seven distance sensors.

the agent must rely solely on the time-varying perceptual feedback that it receives as a result of its actions in the environment in order to successfully perform the task.

In the next section, we describe the model agent and environment that were used in this study and describe the evolutionary protocol that was used to evolve agents. In the third section, we then describe results from a series of experiments exploring the behavior of the best evolved agent. Finally, in the fourth section, we summarize the results from these experiments and then conclude.

## Methods

The model agent used in this study has a circular body with a diameter of 30, and an array of 7 distance sensors equally spaced over an angle of  $\frac{\pi}{3}$  radians on the agent's top side (Figure 1). Each distance sensor has a maximum length of 220. Distance sensors take on values inversely proportional to the distance at which their corresponding rays intersect objects in the environment. This part of the agent model is essentially the same as in previous work on categorical perception (Beer, 1996, 2003; Williams et al., 2008). The agent is positioned along the bottom edge of a planar environment and is able to move horizontally in either direction. The agent's motion is produced by two sets of effectors. One set of effectors, henceforth referred to as *wheels*, propel the agent in either direction with a pure force having a maximum magnitude of 6. The other set of effectors control a simple model leg that the agent can use to walk in either direction. The leg is controlled by three effectors, with two governing left and right swing and the third controlling the position of a foot. When the foot is up, the two swing effectors allow the agent to swing the leg through a range of  $[-\frac{\pi}{4}, +\frac{\pi}{4}]$  with a maximum angular velocity of 5, while the body remains still. If the foot is down, the swing effectors can exert a force to move the agent either left or right. The leg can exert a maximum force of 8 and a maximum torque

of 5 to move the body. When the leg exerts a force on the body, it stretches elastically so that the body's vertical position remains unchanged. However, if the leg reaches either extreme of the allowed angular range of motion, the agent's velocity immediately drops to 0. The agent's motion is also impeded by a constant frictional force of 1, which must be overcome by the effectors in order to produce movements.

The agent's task is to catch circles that fall towards it from above. Specifically, circles of diameter 40 fall towards the agent from an initial vertical distance of 220 (the maximum length of each ray sensor) and at a constant vertical velocity of -1. The agent is to catch each circle by minimizing its horizontal separation from the circle when the circle completes its fall. During evolution, agents were evaluated on 10 circle presentations in each of three motor conditions (explained momentarily), uniformly distributed over a range of horizontal offsets between  $[-150, +150]$  relative to the agent. The agent's performance on each trial is given by:

$$\frac{150 - d}{150}$$

where  $d$  is the distance between the agent and the circle when the circle completes its fall, clipped at 150, and 150 was chosen because it is the maximum initial horizontal offset at which circles are presented.

The agent's performance is evaluated under three different motor conditions. In the first condition, referred to as the *walking and wheels* condition, the agent must coordinate the behavior of both sets of effectors in order to catch the object. This condition can be thought of as the natural behaving state for the agent. In the second *walking only* condition, the agent's wheels effectors are lesioned, such that they have no effect on the agent's motion. In this case, the agent must catch circles using only its leg. Finally, in the third *wheels only* condition, the agent's leg is lesioned, and the agent must use only its wheel effectors to perform the task. Overall performance is then calculated by averaging trial performance for all 10 object offsets in each of the three motor conditions.

The agent's behavior is controlled by a continuous-time recurrent neural network (Beer, 1995b) with the following state equation:

$$\tau_i \dot{s}_i = -s_i + \sum_{j=1}^N w_{ji} \sigma(s_j + \theta_j) + I_i \quad i = 1, \dots, N$$

where  $s$  is the state of each neuron,  $\tau$  is the time constant,  $w_{ji}$  is the strength of the connection from the  $j^{th}$  to the  $i^{th}$  neuron,  $\theta$  is a bias term,  $\sigma(x) = \frac{1}{1+e^{-x}}$  is the standard logistic activation function, and  $I$  represents an external input. The output of a neuron is  $o_i = \sigma(s_i + \theta_i)$ . The agent's sensors are fully connected to a layer of seven interneurons, which are fully interconnected and which project fully to the five motor neurons. In addition, to cut down on the number

of parameters that need to be evolved, the agent's neural architecture is forced to be bilaterally symmetric.

Neural parameters are evolved using a real-valued genetic algorithm with rank based selection. A fitness scaling multiple of 1.01 and a mutation variance of 4 were used. The following parameters, with corresponding ranges, are evolved: time constants  $\in [1, 20]$ , biases  $\in [-16, 16]$ , and connection weights (from sensors to neurons and between neurons)  $\in [-16, 16]$ . Simulations are integrated using the Euler method with a step size of 0.1. In addition, in preliminary evolutionary runs it was discovered that, by evolving agents in all three motor conditions from random initial conditions, agents would converge prematurely to solutions that performed well in the wheels only condition but poorly in the other two conditions. Presumably this finding is due to the fact that walking is a much more difficult behavior to evolve than motion via pure force effectors, and so walking performance was unable to bootstrap itself before the wheels only condition had already been optimized. In order to overcome this difficulty, agents were evolved initially in the walking only condition until an average performance of 90% was reached, and only then were they evolved under all three motor conditions. On the order of 3,000 generations were required to reach an initial level of 90% proficiency in the walking only condition, and then agents were evolved for an additional 10,000 generations in all three motor conditions. A population size of 200 was used in all evolutionary runs.

## Behavioral Analysis

The best evolved agent achieved a mean performance of 97.1% on 5,000 evaluation trials with horizontal offsets uniformly distributed between  $[-150, +150]$  for each of the three motor conditions, with performances of 98.6% with wheels only, 96.5% with walking only, and 96.1% with walking and wheels. The performance of the best evolved agent is shown in Figure 3. From this, it is clear that the agent exhibits a high-performing and general solution to the task. Accordingly, the next question that we would like to ask is how this works. In particular, how does the agent utilize different patterns of feedback to produce the different behaviors? For that matter, how different are the behaviors to begin with? Does the agent's neural circuit use different autonomous dynamics to produce the different behaviors, or is the behavior truly a collective property of the entire brain-body-environment system? While some of these questions are beyond the scope of the present study, we can move towards answering them by performing a detailed analysis of the agent's behavior. By examining the agent's behavior and how it changes under various perturbations, we can begin to constrain the possible underlying mechanisms that might give rise to it.



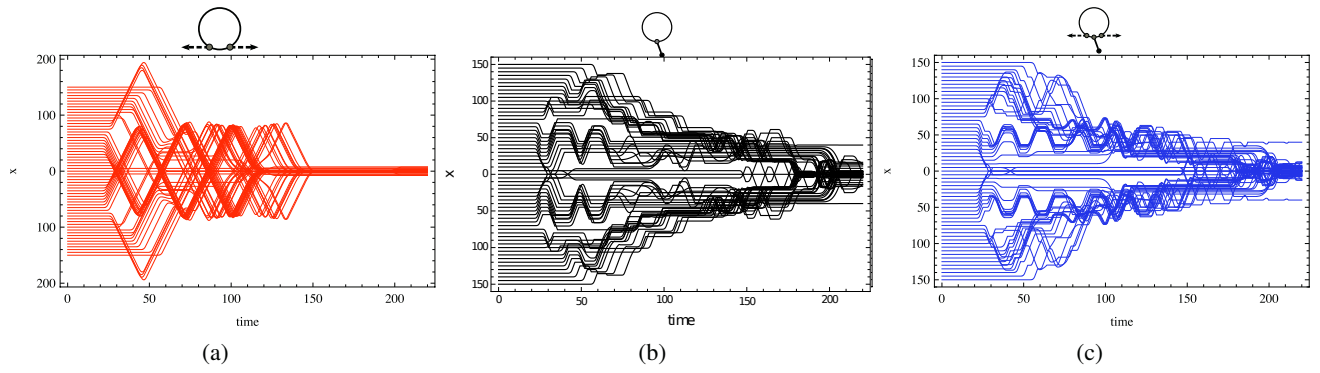


Figure 2: The agent's behavior. The agent's motion over time in object-centered coordinates is shown for the (a) wheels only, (b) walking only, and (c) walking and wheels conditions.

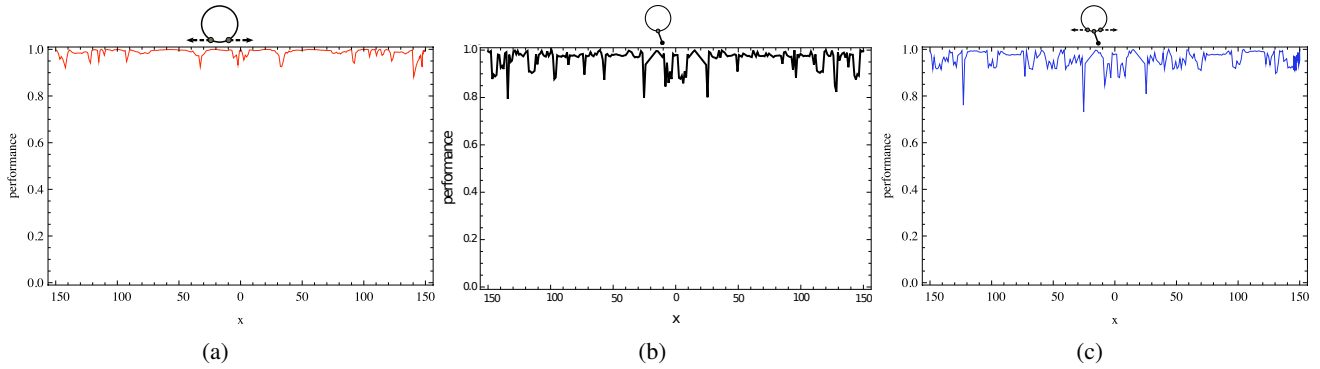


Figure 3: Generalization performance over initial horizontal position for the best-evolved agent. Performance is shown for the (a) wheels only, (b) walking only, and (c) walking and wheels conditions.

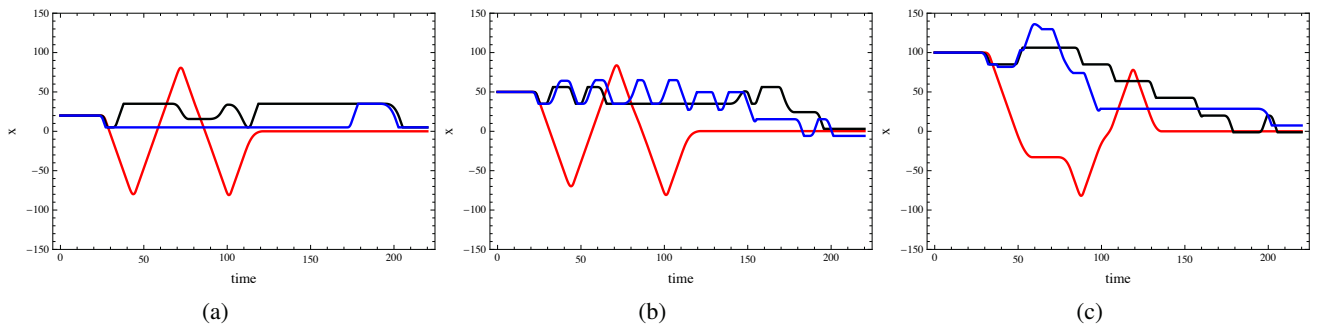


Figure 4: Behavioral comparison. Each plot shows the agent's behavior in the wheels only (red), walking only (yellow), and walking and wheels (blue) conditions when the same stimulus is presented.

## Normal behavior

We start by examining the agent's behavior under normal circumstances. This is shown in Figure 2, where sample trajectories of the agent's motion in each of the three motor conditions are shown. In the wheels only condition, the agent's behavior is characterized by large scans back and forth over the object, before ultimately centering the object as it reaches the bottom of its fall. Interestingly, for offsets

around 100 in the wheels only condition, the agent actually begins by moving further away from the object before turning back and centering it. The agent's motion in the walking only and walking and wheels condition show striking differences from the wheels only condition, largely due to the different biomechanics for walking versus wheels. Successful walking requires that the agent alternate between exerting force while the foot is down and swinging the leg back while the foot is up, resulting in a motion trajectory that al-

ternates between short bursts of motion and stasis. In contrast, with the wheels the agent is able to glide smoothly back and forth unimpeded. The plots in Figure 2 also show apparent similarities between the walking only and the walking and wheels conditions, giving an initial impression that behavior in these two conditions is more closely related than either is to the wheels only condition.

### Behavioral comparison

Next we can ask how much the agent's behavior actually differs in the three motor conditions. Since it is a basic supposition of this paper that the neural circuit exhibits *different* behaviors, this is an important question to ask. Here a clarification also must be made regarding the intended meaning of "behavior". At one level, the agent's behavior can be considered the same in all three conditions, since in each case the agent realizes the same high-level goal of catching circles. However, if behavior is defined at the lower level of the actual motor trajectories that are produced, then the behaviors may in fact be different in each case. That is, although the agent realizes the same goal in each condition, the actual motor actions required for walking versus wheels versus coordinating both may differ substantially. To determine whether this is the case, we can begin by comparing the agent's behavior in the three conditions when presented with the exact same stimuli (Figure 4). From this comparison, we see that, despite the apparent similarities between the walking only and walking and wheels conditions in Figure 2, the behavior in all three conditions is actually quite different. The contrast is most clear in Figure 4(c), where the trajectory for the walking and wheels condition can be seen to combine the steplike motion of the walking condition with periodic glides characteristic of the wheels only condition.

While Figure 4 provides an initial qualitative comparison of behavior in the three conditions, it is also possible to perform a more rigorous and quantitative comparison. To do this, we record the agent's motor outputs in one of the three conditions and then "playback" various of the motor streams to determine how the agent would have performed in the other conditions. For example, when the agent is evaluated in the wheels only condition, the agent actually produces motor outputs for the leg as well, but those outputs are simply ignored in order to simulate the leg being lesioned. Thus, if we record the outputs of the leg motors during the wheels only condition and then subsequently play them back, using them to drive the agent's leg effectors, we can examine how the agent would have performed had it been in the walking only condition. Similarly, we can play back the motor streams for both the wheels and leg to simulate the walking and wheels condition. In general, we can perform the same experiment by running the agent in each of the three conditions and performing playback simulations of the other two. The results from performing these experiments are shown in

Figure 5, both as average performances and as performance across the range of horizontal offsets. Clearly, there are significant drops in performance in all of the playback conditions. The largest drops are found between the wheels only condition and the other two, in line with our earlier observation that behavior is most different in the wheels only condition. However, even between the walking only and walking and wheels conditions there are significant declines in performance. Thus, the results of these experiments strongly support our earlier qualitative observations that the agent does in fact produce different behaviors in each of the three conditions.

### Effects of removing the object

Having established that the agent exhibits different behaviors, we can next ask about the source of this behavioral differentiation. In particular, to what extent does the differentiation rely on continuous feedback from the environment? If we found, for instance, that the agent does not rely on continuous feedback, this would suggest that the agent's neural circuit may be intrinsically multifunctional, with initial environmental input serving only to switch the agent into one or the other of its behavioral modes. On the other hand, if the different behaviors rely on continuous feedback, this would lend support to the idea that the environmental feedback is in fact crucial for producing the different behaviors. To determine which is the case, we can remove the visual object at different times during each trial and measure the impact on performance. Figure 6 shows the results of performing these experiments. Performance in all three conditions is significantly impaired by removing the object at nearly all times except very late in the trial, presumably after the agent has already settled on its final position. Also, interestingly, whereas the walking condition shows a steady increase in performance as the object is removed later in the trial, the other two conditions show much greater variability. There are certain times when the agent is very sensitive to the object being removed, and certain other times when performance is hardly affected at all. Also, by comparing the density plots in Figure 6 with the behavioral trajectories in Figure 2, one can begin to see why this is likely the case. The points in time when performance is impacted the most appear to correspond to times when the agent's behavioral trajectory is changing, turning either towards or away from the object. Thus, one reasonable prediction is that environmental feedback influences behavior precisely at these critical junctures, when the agent is actively moving to position itself with respect to the object.

### Effects of switching motor conditions

The final set of experiments examine the agent's ability to flexibly switch between the motor conditions. There are several reasons why the results of these experiments are of interest. First, the ability to switch between conditions provides a

measure of the robustness of the agent's control mechanism. Since, as is well established by now, the behavioral trajectories in each condition are actually quite different, it is not at all obvious that the agent should be able to switch behaviors mid-trial. Successful walking, for example, may rely fundamentally on the precise pattern of feedback that the agent selects for itself while walking, which differs significantly from feedback in the walking and wheels condition. Thus, to the extent that the agent is able to switch conditions, we can explore the robustness of the evolved mechanism. A second reason that these experiments are of interest is that the ability to switch behaviors would also provide further support for the idea that the agent's behavior relies fundamentally on continuous environmental feedback. For example, if the agent uses feedback only to switch into one or the other behavioral mode, and ignores it thereafter, then the agent presumably will fail when switched to a different condition. On the other hand, if the agent is continuously adjusting its behavior online as a result of changing patterns of feedback, then it is more likely able to adapt to a different motor condition.

The results of switching motor conditions at different times during the trial are shown in Figure 7. First, we see that when switching from the wheels only condition to either of the other two conditions performance decreases significantly, especially as the switch occurs later in the trial. One possible explanation for this is that the agent sweeps back and forth over the object much more widely in the wheels

only condition, and when switched to the walking only or walking and wheels condition it may be unable to recover this distance before the object completes its fall. However, when switching from either of the other two conditions, performance remains high regardless of when the switch occurs. This is a somewhat surprising result, especially considering switches to the wheels only condition which, as we have seen, involves very different behavior. Moreover, this result also provides strong support for the earlier findings of Section 3.3, indicating that the agent's behavior relies fundamentally on continuous environmental feedback.

## Discussion

Although neural mechanisms are undoubtedly crucial in producing different behaviors, the embodied and embedded contexts of neural circuits also provide many additional degrees of freedom that are often under-appreciated. Behavior is the product not of brains, but of entire brain-body-environment systems, and each of these three components may have a profound influence on behavior. This paper explored the ability of environmental feedback to drive the production of different behaviors from a single fixed neural circuit. Agents were evolved to accomplish the same objective—catch circles—under three different motor conditions, and based solely on the different patterns of environmental feedback produced by these conditions. The successful evolution of agents in this task demonstrated the ability of environmental feedback alone to drive behavioral selec-

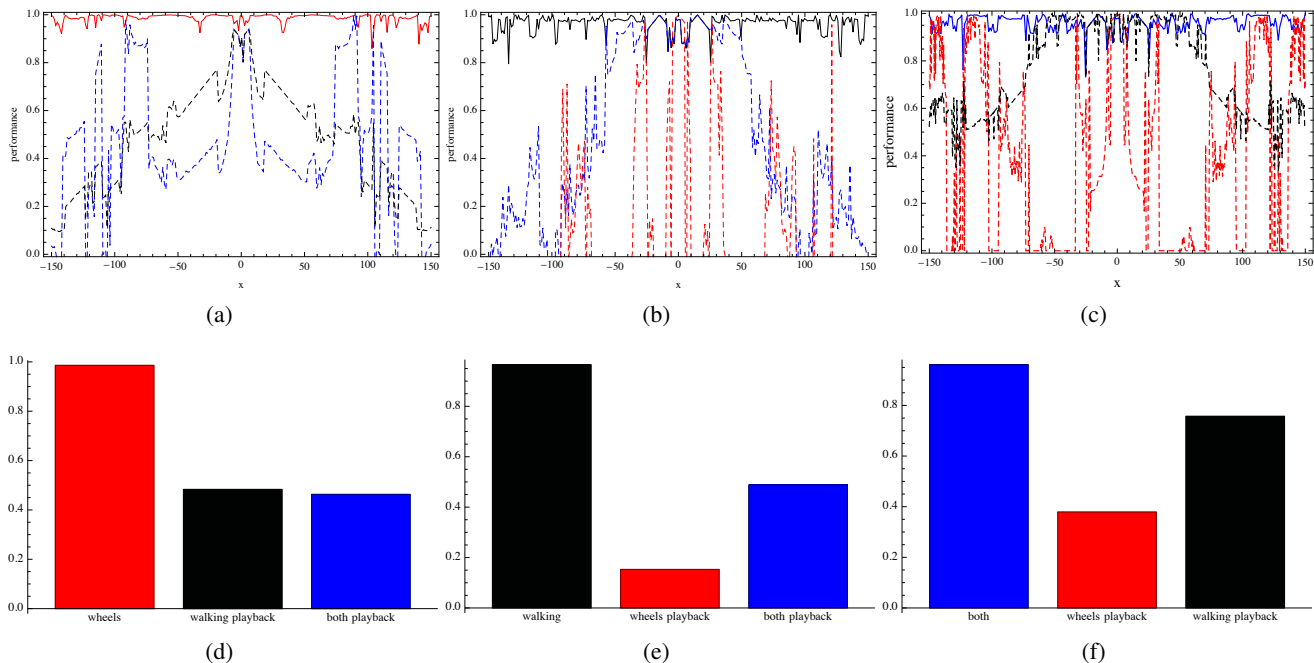


Figure 5: Playback performance. Plots (a)-(c) show the agent's normal performance (solid line) and playback performance (dashed lines) for the wheels only (red), walking only (yellow), and walking and wheels (blue) conditions. Average performances for the normal and playback conditions are shown in plots (d)-(f).

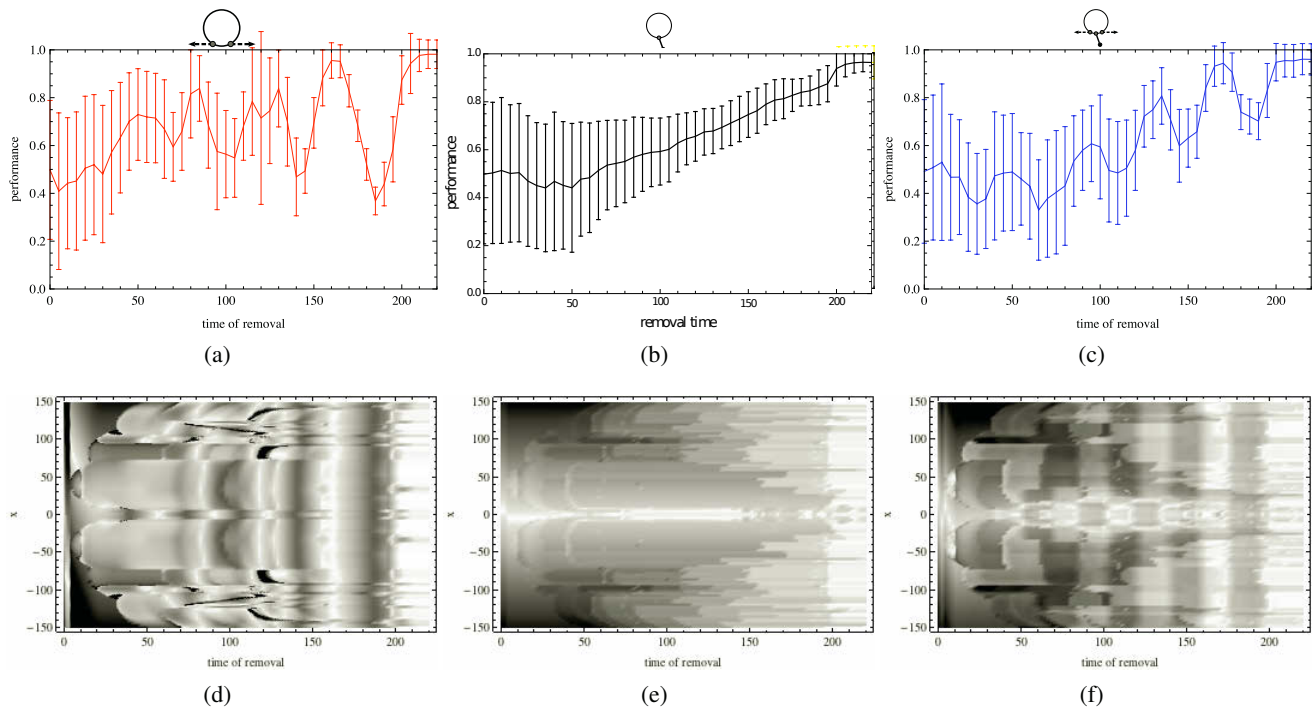


Figure 6: The effect of removing the object. The top row shows the agent’s average performance as a function of the time when the object is removed, averaged over horizontal positions of the object. The bottom row shows density plots of performance as a function of the object’s horizontal position and the time of removal.

tion. Next, a detailed analysis of the behavior of the best evolved agent was performed. Preliminary experiments provided quantitative evidence in support of the claim that the behaviors produced by the agent differ significantly. Next, experiments where the agent’s environmental feedback was removed at different times during each trial, by removing the object that the agent is supposed to catch, showed the agent’s fundamental reliance of continuous environmental feedback. Finally, experiments where the agent was switched between motor conditions at different times demonstrated the robust behavioral mechanism that the agent employs, and further confirmed the agent’s reliance on continuous environmental feedback.

### Acknowledgements

This work was supported by NSF grants IIS-0916409 and IIC-1216739.

### References

- Auerbach, J. and Bongard, J. C. (2009). How robot morphology and training order affect the learning of multiple behaviors. *2009 IEEE Congress on Evolutionary Computation*.
- Beer, R. (1995a). A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 72:173–215.
- Beer, R., Chiel, H., and Gallagher, J. (1999). Evolution and analysis of model cpgs for walking: II. general principles and in-

dividual variability. *Journal of Computational Neuroscience*, 7:119–147.

- Beer, R. D. (1995b). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3(4):469–509.
- Beer, R. D. (1996). Toward the evolution of dynamical neural networks for minimally cognitive behavior. *From Animals to Animats 4*, pages 421–429.
- Beer, R. D. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11(4):209–243.
- Beer, R. D. (2008). The dynamics of brain-body-environment systems: A status report. In Calvo, P. and Gomila, A., editors, *Handbook of Cognitive Science: An Embodied Approach*, pages 99–120. Elsevier.
- Beer, R. D. and Gallagher, J. (1992). Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1(1):91–122.
- Briggman, K. L. and Kristan, W. B. (2008). Multifunctional pattern-generating circuits. *Annual Review of Neuroscience*, 31:271–294.
- Brooks, R. (1991). New approaches to robotics. *Science*, 253:1227–1232.
- Buhrmann, T., Di Paolo, E. A., and Barandiaran, X. (2013). A dynamical systems account of sensorimotor contingencies. *Frontiers in Psychology*, 4.



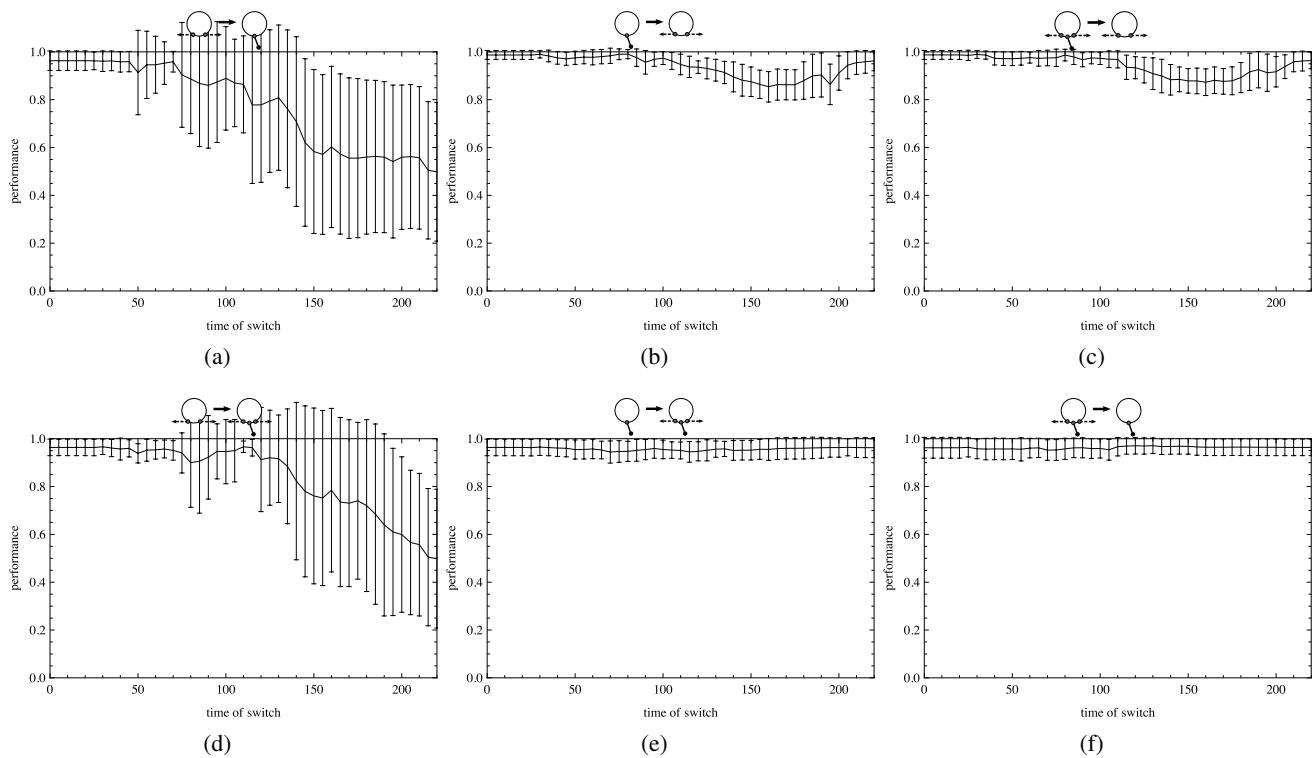


Figure 7: The effect of switching motor conditions. The agent's average performance as a function of the time of switch is shown for the following switches: wheels only to (a) walking only and (d) walking and wheels; walking only to (b) wheels only and (e) walking and wheels; and walking and wheels to (c) wheels only and (f) walking only.

Clark, A. (1995). *Being There: Putting Brain, Body, and Environment Together Again*. MIT Press.

Fine, P., Di Paolo, E. A., and Izquierdo, E. (2007). Adapting to your body. *Advances in Artificial Life: Proceedings of the Ninth European Conference on Artificial Life*, pages 203–212.

Getting, P. A. (1989). Emerging principles governing the operation of neural networks. *Annual Review of Neuroscience*, 12:185–204.

Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA: The MIT Press.

Izquierdo, E. and Buhrmann, T. (2008). Analysis of a dynamical recurrent neural network evolved for two qualitatively different tasks: walking and chemotaxis. *ALife XI*.

Kirsh, D. and Maglio, P. (1994). On distinguishing epistemic from pragmatic action. *Cognitive Science*, 18(4):513–549.

Lungarella, M. and Sporns, O. (2005). Information Self-Structuring: Key Principle for Learning and Development. *Development and Learning, 2005. Proceedings. The 4th International Conference on*, pages 25–30.

Morton, D. W. and Chiel, H. J. (1994). Neural architectures for adaptive behavior. *TINS*, 17(10).

Nolfi, S. (2009). Behavior and cognition as a complex adaptive system: Insights from robotic experiments. In Hooker, C., editor, *Handbook of the Philosophy of Science: Philosophy of Complex Systems*, volume 10. Elsevier.

Pfeifer, R. and Bongard, J. (2007). *How the Body Shapes the Way We Think*. MIT Press.

Pfeifer, R., Lungarella, M., Sporns, O., and Kuniyoshi, Y. (2007). On the information theoretic implications of embodiment—principles and methods. *LNAI*, 4850:76–86.

Polani, D., Sporns, O., and Lungarella, M. (2007). How information and embodiment shape intelligent information processing. *LNAI*, 4850:99–111.

Williams, P. L., Beer, R. D., and Gasser, M. (2008). An embodied dynamical approach to relational categorization. *CogSci 2008*, pages 223–228.

## A-Bees See: A Simulation to Assess Social Bee Visual Attention During Complex Search Tasks

Zoe Bukovac<sup>1</sup>, Alan Dorin<sup>1</sup> and Adrian G. Dyer<sup>1,2</sup>

<sup>1</sup> Monash University, Clayton, Australia 3800

<sup>2</sup> RMIT University, Melbourne, Australia 3000

zoe.bukovac / alan.dorin @monash.edu (corresponding author), adrian.dyer@rmit.edu.au

### Abstract

Foraging bees often search in complex natural environments for "target" flowers that they have learnt provide nectar rewards. To maximise efficiency, bees must avoid landing on "distractor" flowers that do not offer rewards, as this potentially wastes time and energy. This paper reports on artificial-life inspired agent-based simulations of two contrasting approaches different bee species use to scan for targets in a scene containing many flowers. The two scanning approaches simulated are a parallel scan typical of bumblebees that is not slowed by distractors, and a serial scan typical of honeybees that is faster than parallel scan for single element processing, but is slowed by the presence of distractor flowers. The simulations were conducted over a range of target densities, and over a range of target/distractor ratios, to evaluate the types of environment in which each scan mechanism is most effective. Serial scan was found to be generally more effective in environments populated with a single type of rewarding species of flower, and parallel scan appears to be relatively more effective in environments populated with a mix of rewarding and unrewarding flowers. Our results support the hypothesis that environmental factors led to the evolution of different visual processing mechanisms in honeybees and bumblebees. This establishes a firm basis for psychophysical research exploring how and why the two different processing mechanisms may have evolved in these animals.

### Introduction and Previous Work

Foraging worker bees collect nutrition to sustain a beehive. Individual bees travel on "bouts" between their hive and flowers that may present nectar and pollen as nutritional rewards. Some bee species like honeybees (*Apis mellifera*) have colonies of foragers numbering in the thousands; while other species like bumblebees (*Bombus terrestris*) typically have less than 100 foragers (Frisch 1967, p.7; Duchateau and Velthuis 1988). Differences in search strategy between individual bees are amplified many times by the numerous bouts an individual travels during a day, many more times depending on the number of bees in a colony sharing that strategy, and more times still over a season or the life of a colony. The evolutionary relationship of different bee species and flowers is likely to have endured over many millions of years (Dyer, Boyd-Gerny *et al.* 2012), suggesting that selective pressure to evolve optimal solutions may exist in current day populations (e.g. examples throughout (Lythgoe

1979)). The search strategy employed should therefore be adapted to local environmental conditions, maximising the net flow of energy into the hive to enable survival and reproductive success of the queen bee. Costs associated with unnecessary workers, or excessive flight and flower handling should be avoided, but this is not straightforward in complex natural environments (Burns and Dyer 2008).

During a foraging bout, bees search for *target* flowers that they have learnt provide nectar rewards. Many social bees, like honeybees and bumblebees, exhibit flower constancy and tend to forage consistently from one type of rewarding flower as long as it continues to present rewards (Chittka, Thomson *et al.* 1999). However, in complex environments with many flowers, bees must avoid landing on unrewarding *distractor* flowers as this wastes time and energy (Burns and Dyer 2008). When the colours of target and distractor flowers are very different (e.g. blue and yellow as seen by humans), and there are only two flowers to choose between, bees can accurately assess the type of a flower presented to them (Giurfa 2004; Dyer, Spaethe *et al.* 2008). However, it isn't currently well understood how bees make decisions in complex environments containing many flowers of different types, colours, target/distractor ratios and arrangements. For instance, in tropical forests, single trees with thousands of simultaneously blooming rewarding flowers, potential targets, may appear (Fig. 1a), and distractor flowers are not intermingled among them (Clark 1994). However, in temperate environments, diverse carpets of small herbaceous plants may include a scattered few targets peppered among numerous distractor species (Fig. 1b), or a uniform carpet of a single species may occur that is many meters across depending upon location or season (Fig. 1c).

Apart from the obvious biological interest foraging raises (Pyke 1984), understanding bee foraging has many practical implications for agriculture where globally, crop bee-pollination is directly responsible for 35% of worldwide food production and is worth an estimated 153 billion Euro annually (Kjøl, Nielsen *et al.* 2011, pp.1-49). Bee pollination is also essential for natural ecosystem management (Hegland, Nielsen *et al.* 2009). It is therefore imperative that we understand how different bee species operate in different environments. This also provides insight into how potential

temporal or spatial mismatches in bee-pollinator and/or flower blooming may be affected by predicted changes in environmental conditions (*ibid.*). Furthermore, understanding bee visual attention mechanisms is relevant to artificial vision and search system design (Srinivasan 2011).

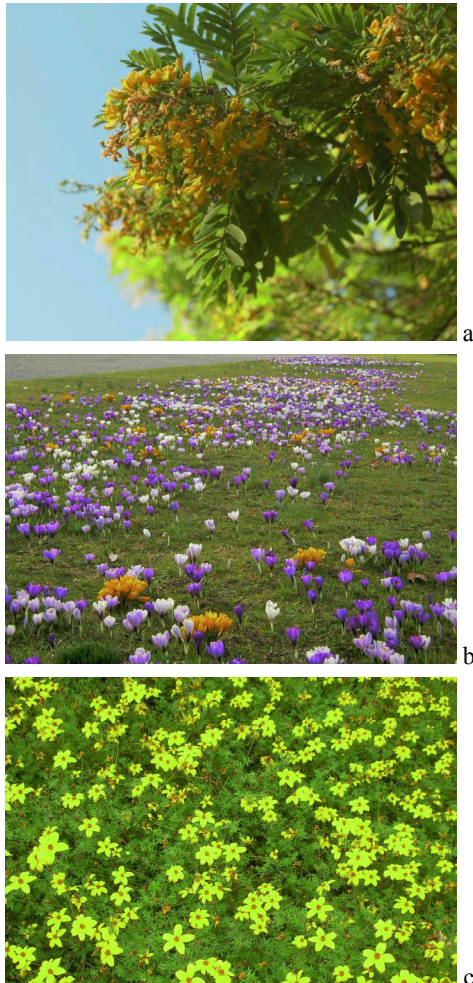


Figure 1: Sample flower distributions: (a) flowering tree, (b) patchy temperate carpet, (c) uniform temperate carpet.

**Biological background.** Following recent experiments with live free-flying bees, Morawetz and Spaethe (2012) propose that bumblebees (*Bombus terrestris*) conduct what appears to be a limited *parallel* style visual scan for targets that is not significantly slowed by increasing the number of distractors, while honeybees (*Apis mellifera*) use a *serial* styled scan that is faster for decisions where only targets are present, but is slowed considerably when distractors that must also be processed by the visual system are present. These different spatial attention mechanisms are potentially required to allow an organism with finite information processing capacity to handle the essentially infinite complexity of applying vision in natural environments (Treisman and Gelade 1980; Treisman and Gormican 1988; VanRullen, Carlson *et al.* 2007).

The study on free-flying honeybees and bumblebees (Morawetz and Spaethe 2012) used differently coloured paper

targets known to stimulate the trichromatic (UV, blue and green sensitive (Dyer, Paulk *et al.* 2011)) colour processing system of bees. This type of behavioral testing relies on the fact that individual bees can be trained to visit a target colour by associating a sucrose solution which the bees collect as nutrition, whilst the distractor stimuli offer no reward to bees who therefore avoid this colour. This scenario is biologically relevant since flowers may only offer rewards in certain temporal cycles, and there are some that mimic rewarding flowers and try to obtain pollination through deception (Dafni 1984; Dyer, Paulk *et al.* 2011). The Morawetz and Spaethe (2012) study was conducted in a controlled arena with stimuli presented at set visual angles. The number of targets and distractors was systematically varied. The results showed that increasing the number of distractors led to a significant increase in the decision-making time for honeybees, consistent with theories of a serial search mechanism (Treisman and Gelade 1980; Treisman and Gormican 1988; VanRullen, Carlson *et al.* 2007). However, in bumblebees a different processing system was observed. While decisions for finding a target with only a single distractor were about 1.5 times as long as for honeybees, increasing the number of distractors did not significantly affect the decision-making time of bumblebees. This type of decision-making is consistent with a parallel visual search (*ibid.*). In the current study we use simulations to test the implications on nectar gathering effectiveness, as a biologically relevant measure of fitness, for parallel and serial scanning mechanisms.

**Simulation background.** Experiments with free-flying bees require marking and tracking individual animals moving freely in 3D space, making it difficult to collect sufficient reliable data to answer iterative questions about optimal mechanisms in multiple environments. Hence, we employ an agent-based model (ABM, or individual-based model, IBM) simulating parallel and serial scanning bees in different environments. Our artificial bees (a-bees) search a grid world populated by target and distractor flowers. We systematically sweep through a biologically relevant range of target densities and target/distractor relative abundances, aiming to determine the environmental floral distributions in which each visual scan mechanism is likely to be effective in real world scenarios. The simulations allow us to interpret the factors that influence how and why bees make decisions, and the subsequent colony-level benefits that may act as biologically relevant factors for reproductive success (Burns 2005; Burns and Dyer 2008).

Where bee behaviour varies between individuals or where local environmental conditions influence individual decision-making, ABMs offer a powerful approach for understanding the intricate interactions and emergent outcomes of these complex systems (Huston, DeAngelis *et al.* 1988; Judson 1994; Grimm 1999; DeAngelis and Mooij 2005; Grimm and Railsback 2005; Grimm, Revilla *et al.* 2005; Dorin, Korb *et al.* 2008). ABMs have been used to model bee behaviour since the 1980s (Hogeweg and Hesper 1983). For example, they have been used to understand bee foraging strategies in keeping with empirical data whilst considering recruitment,



homing and memory of food source location (Vries and Biesmeijer 1998); and to show that the benefits of recruitment by honeybees is dependent on the density of flowers within certain environments (Dornhaus, Klügl *et al.* 2006). ABMs have also been applied to understand the impact of flower constancy under conditions where flower rewards become available or unavailable cyclically (Dyer, Dorin *et al.* 2012).

In this current study we use ABMs to understand the potential colony level benefits of the empirically determined visual scanning behaviour discovered in honeybees and bumblebees. Specifically, we hypothesise that if environmental conditions like flower density have been a factor in the evolution of different bee species' visual search mechanisms (Dyer, Spaethe *et al.* 2008; Morawetz and Spaethe 2012), then there should be evidence of biologically plausible flower distributions best suited to foraging by bees employing the different mechanisms.

## Simulation Method

The ABM we have designed simulates the components pertinent to testing the hypothesis just stated, while eliminating irrelevant factors – KISS.<sup>1</sup> In detailing the simulation here we discuss aspects of real bee behavior included or excluded to test our hypotheses. The system was implemented entirely in the C programming language but there is nothing in our model that could not be built in any other similar language by following the description below.

### Artificial-Bees (A-Bees)

Bees are central place feeders that leave their hive to search for nutrition. We model bees as software agents, a-bees, foraging within a virtual bounded foraging patch. The patch is uniformly divided into square grid cells. At most one virtual flower can occupy a grid cell. An a-bee also occupies a single grid cell with or without a flower. In this section we detail a-bee behaviour, the simulated environment and the relationship of these aspects of our model to reality.

We model bee colonies of 60 foragers. The exact number is not critical since we eliminate inter-agent communication and population density effects by ensuring that, in essence, each a-bee exists in a world of its own. Inter-agent effects are complicating factors that would change the viability of different visual scan mechanisms for hives of different size and under different environmental conditions. In keeping with KISS described above, the simulation eliminates these to reduce one key problem to its basic form.

Honeybees have evolved a complex language for communicating target whereabouts to one another (Frisch 1967, pp.321-328). This is of particular benefit in environments where targets appear in large clusters (Dornhaus, Klügl *et al.* 2006). This communication system is likely to impact on the effectiveness of different bee species'

visual scanning techniques, but we have avoided introducing it here in order to establish a baseline for comparing only the visual scans of honeybees against those of bumblebees (Dornhaus and Chittka 1999).

Bees can use vision and olfaction to help find flowers; e.g. (Streinzer, Paulus *et al.* 2009). Our model only considers visual scan. Our a-bees can distinguish between targets and distractors with 100% accuracy. This is biologically plausible for saliently different colours (Giurfa 2004; Dyer, Paulk *et al.* 2011).

Bee spatial acuity is relatively poor compared to human vision, and in real life bumblebees can only detect a plant's cluster of 3-5 flowers (each flower of 2.5cm diameter) at a distance of approximately 0.7m (Dyer, Spaethe *et al.* 2008; Wertlen, Niggebrugge *et al.* 2008). Detection appears to approximate a step function, so we model it as distances >0.7m — not detected, distance <0.7m — 100% chance of detecting a flower that is present. Our foraging patch model is based on a grid world of square cells of dimension 0.35m, so an a-bee can see flowers in its Moore neighbourhood (8+1 cells, n=1), but no further.

An individual a-bee keeps track of the flowers it has visited and will not visit a flower twice in a single foraging simulation. This is biologically plausible and the ability is an important aspect of bee foraging behavior (Giurfa, Núñez *et al.* 1994). An a-bee will move into a grid cell towards a target if it has not visited that target before. It will then visit that target (to collect the modeled nutritional reward), taking a parameterised amount of time, *VisitTime*. Real bees do have a central foveal region in which they may see detail better. However its operation in complex environments for different species has been poorly studied (but see (Morawetz and Spaethe 2012)). For simplicity we eliminated this factor here; a-bees are not directionally biased in their visual scan.<sup>2</sup> We model the two bee scan mechanisms as follows.

**Parallel scan.** An a-bee using a parallel scan (Fig. 2) processes all of the information it sees about flower locations in its visible range simultaneously. This process takes the "parallel a-bee" a constant amount of time, *ParallelTime*, regardless of how many flowers it can see and regardless of whether these are distractors or targets. It is as if in *ParallelTime* the bee forms a mental image of the whole visible scene and recognises the closest target, while ignoring all non-targets.

A parallel a-bee will move towards an unvisited target it sees that is drawn from a uniform random distribution of available target flowers. If it finds no target, or no flowers at all, it will conduct a random walk as discussed shortly.

<sup>2</sup> However, we did test the impact of bias in the a-bees' preferred direction of travel (below).

<sup>1</sup> Keep It Simple, Stupid (Axelrod 2003).



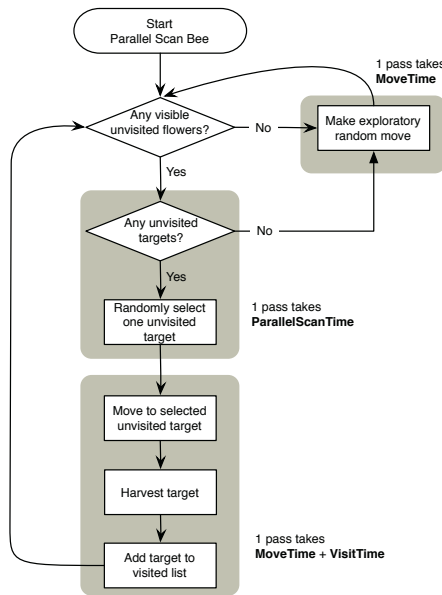


Figure 2: The foraging cycle of a parallel visual scan a-bee.

**Serial scan.** An a-bee using a serial scan (Fig. 3) examines one flower in its visible region at a time, selecting flowers to investigate from a uniform random distribution across available flowers. It takes this “serial a-bee” the parameter value *SerialTime* to examine *each* flower, until it finds a target, or until it has examined all the flowers it can see and finds no targets. A serial a-bee always moves towards the first unvisited target it sees in its scan. It stops scanning as soon as it finds an unvisited target. In the absence of unvisited targets or any flowers at all, the serial a-bee conducts a random walk as discussed below.

**A-bee movement.** As long as all a-bees, regardless of parallel/serial visual scanning mechanism, apply the same movement strategies, we can assess the *relative* benefits of the scanning mechanisms free from interference introduced by movement strategies that are themselves complex and worth independent study (Waddington 1980). Hence it is especially important here to eliminate unnecessary complications from the simulation. Since we are concerned with bee decision-making times and unconcerned about bee travel times, we don’t require a-bees to return to a hive and therefore don’t model one. A-bees do not run out of nectar storage; they accrue it indefinitely during a simulation run.

In parameter *MoveTime* simulation time-steps an a-bee can move into any cell in its Moore neighbourhood, or choose to remain in its current cell. But in which direction should it head? If it sees a target flower, as discussed above, it will move towards that. In the absence of target flowers, a-bees conduct either a random walk or a *biased* random walk around their Moore neighbourhood depending on the experiment.

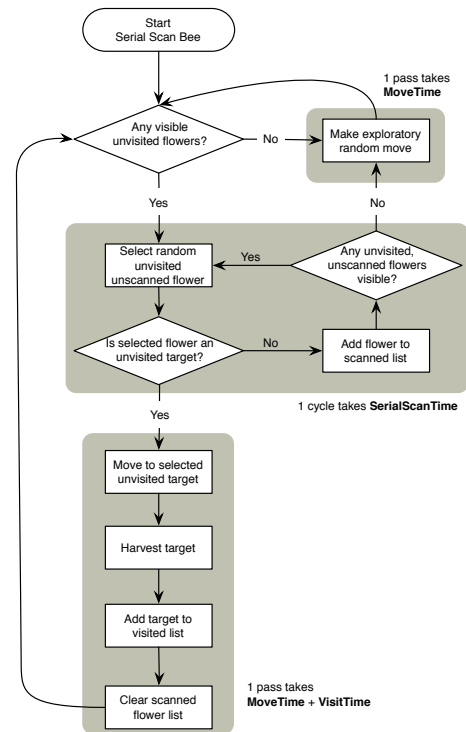


Figure 3: The foraging cycle of a serial visual scan a-bee.

While an ordinary random walk between grid cells is not a biologically realistic foraging strategy, it provides a convenient baseline against which we compare the impact of a *biased* random walk derived from a study of real bees’ directional preferences (Waddington 1980, Fig. 1). The probability of a biased random walking a-bee selecting a specific cell from its Moore neighbours in the absence of suitable target flowers is given in Fig. 4. As can be seen, it prefers to continue straight ahead, but is not completely averse to changing direction. We investigated these two navigational strategies to determine if they had any impact on the relative success of the visual scan techniques.

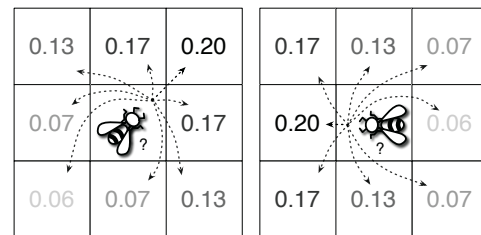


Figure 4: Probability of movement relative to (example) current headings of an a-bee following a biased random walk. Values calculated from (Waddington 1980, Fig. 1).

### Artificial Foraging Environments

A colony’s foraging environment is modeled as a bounded world of  $571 \times 571$  square grid cells. Cells represent  $0.35\text{m}^2$ . Hence, the simulated foraging site is 200m across.

Target flowers offer a reward of 1 unit to every a-bee that visits them. Distractor flowers offer no reward.<sup>3</sup> Flowers are distributed differently across the patch depending on the experiment, as described in the next section.

## Experiments

We conducted experiments to determine the impact of various flower distributions on the relative success of colonies composed either entirely of parallel a-bees, or entirely of serial a-bees. The simulation parameters described below are summarised in Table 1.

**Target-only experiments.** In some environments, bees only encounter a single flowering species during a foraging bout (Fig. 1a & 1c). This is often the case under tropical conditions where massive flowering trees can be encountered, or in homogeneous agricultural fields. In experiments conducted to model these conditions, colonies of serial and parallel a-bees foraged in patches containing only targets. The number of target flowers in the patch was increased systematically to occupy from 0% to 100% of the grid cells. Serial and parallel visual scanning mechanisms were compared at each density. Flower positions were distributed uniform randomly across the patch in each case. These experiments were primarily used to inform the choice of target density to be used in the *Target and distractor experiments* described next.

**Target and distractor experiments.** In order to ascertain the relative impact of distractor flowers on the serial and parallel visual scanning mechanisms, distractors were placed in the foraging patch among a fixed number of targets. In our simulation, distractor flowers form a single class of non-target flower. In the wild there may be many species of distractor. However, our simulation remains biologically relevant since, as detailed in section Biological background, when distractor flowers are of saliently different colour to targets, bees reliably distinguish between them.

For the Target and distractor experiments, the density of target flowers in our simulated patch was fixed at 10% of the grid cells. There were three reasons for this density choice:

- (i) Since we are measuring the relative success of the two visual scanning mechanisms by recording the total number of reward units gathered during the simulations, we always require some target flowers for a-bees to harvest.
- (ii) To maximise our ability to distinguish differences between visual scan mechanisms we need to provide sufficient targets that foraging a-bees do not exhaust the supply of unvisited targets during a simulation run.
- (iii) We need to sweep across a wide range of distractor flower densities. By fixing targets at 10% we have from 0% up to the remaining 90% of grid cells to populate with distractors.

<sup>3</sup> Actually, a-bees never visit distractors as they distinguish flower types with 100% accuracy. This is consistent with empirical data for bees visiting saliently different coloured flowers (Giurfa 2004).

Our Target-only experiment results demonstrated that 10% target density met all three requirements. These results are described below. Hence, during Target and distractor experiments, the number of distractors in the patch was increased systematically to occupy from 0% to 90% of the grid cells. The two visual scanning mechanisms were tested at each distractor density. Flowers were distributed uniform randomly across the patch in each case, simulating a temperate environment with different numbers of distractors positioned among targets (Fig. 1b).

Environment	
Patch size	571 × 571 cells, bounded
Patch grid cell size	0.35 × 0.35m
Colony size	60 parallel, or 60 serial scan a-bees
A-bees	
Flower presence detection accuracy	100% from neighbouring cell or cell shared with a flower
Flower type recognition accuracy	100% from neighbouring cell or cell shared with a flower
Storage capacity	Infinite
Visited flower memory length	Every flower visited in a simulation run
Flower visit ( <i>VisitTime</i> )	1 simulation time step
Complete field of view parallel scan ( <i>ParallelTime</i> )	3 simulation time steps
Single serial scan flower examination ( <i>SerialTime</i> )	2 simulation time steps
Movement in Moore neighbourhood ( <i>MoveTime</i> )	1 simulation time step
Simulation	
Duration	1000 simulation time steps
Number of runs	20 per data point

Table 1: Main simulation parameters.

## Simulation verification

We tested that the simulation behaved according to the specifications above. Tests included that a-bees were:

- Correctly following their respective visual scanning mechanisms in assessing visible flowers;
- Remembering visited target flowers;
- Not exhausting the global target flower supply;
- Not exhausting the local target flower supply.

Where possible we also compared analytically derived values to simulation results. To ensure our a-bees' simplistic random walk navigation strategy did not influence the *relative* success of the tested scanning mechanisms, we compared the results of this navigation strategy against a more plausible biased walk derived from empirical data. See the Results for a discussion of these experiments.

## Results

### Target only experiments

In these experiments the foraging patch was filled purely with target flowers. The number of target flowers was increased systematically from 10% to 100% of the grid cells in the patch. The effects of this increase on the amount of nectar collected are shown for colonies of parallel and serial a-bees (Fig. 5). All bees in these experiments executed an ordinary random walk.

At target densities of 1 or 2 % there was a relatively low rate of nectar collection by a-bees with either serial or parallel search mechanisms (Fig. 5). A comparison of nectar collection success for colonies of a-bees using parallel (mean  $4920 \pm 103$  units) or serial (mean  $5323 \pm 118$  units) visual search considering 10% target density and no distractors was conducted with a non-parametric Mann-Whitney U test (SPSS v15.0: IBM, Chicago, USA) for 2-independent groups of  $N=20$  runs/group ( $Z=-5.383$ ,  $p<0.001$ ). Thus at 10% target density, a-bees with a serial search mechanism collected significantly more nutrition. As target flower density increases in the presence of no distractors, the serial search becomes increasingly more effective than the parallel search mechanism (Fig. 5).

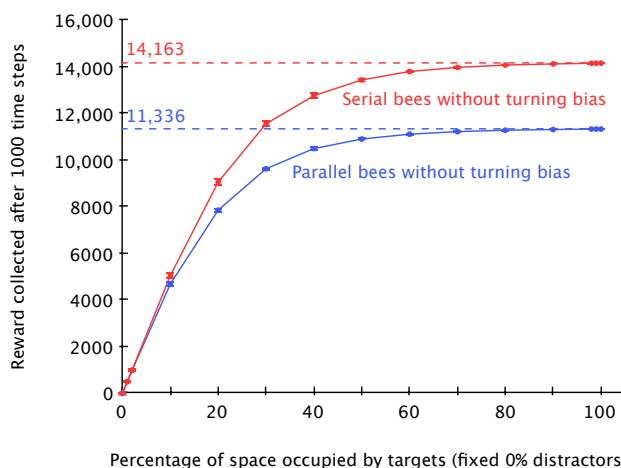


Figure 5: Graph of nectar rewards collected by colonies of a-bees versus the percentage of grid cells occupied by target flowers. Data points are the mean of 20 simulations; error bars  $\pm 1$  std. dev.

### Target and distractor experiments

In these experiments 10% of the foraging patch was randomly filled with targets. Distractor density was increased systematically from 0% to 90% of the total number of grid squares. The distractors were randomly distributed among the cells unoccupied by targets. The effects of this increase on the amount of nectar collected by colonies of parallel and serial a-bees executing ordinary random walks and biased random walks are shown (Fig. 6).

Considering first the a-bees executing an ordinary random walk. For distractor densities less than 1%, the serial scan mechanism outperforms the parallel scan (see Fig. 5 above,

since the Target only experiment at 0% distractors, matches the experiment here at 0% distractors). However, parallel scan takes over as the most efficient mechanism at higher distractor densities, even where the number of distractors is much less than the number of targets. Parallel scan is clearly more efficient than the serial scan as distractor flower density increases beyond target density.

The trends for a-bees conducting a biased random walk correspond directly to those for a-bees conducting an ordinary random walk, but with greater overall success for the former over the latter. The directional bias appears to enhance the speed with which a-bees' locate unvisited targets. It does this in equal proportion for a-bees using parallel or serial visual scan and didn't change the relative success of these mechanisms.

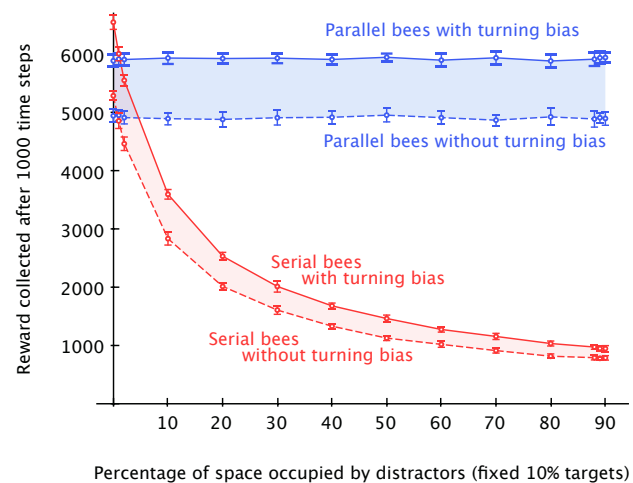


Figure 6: Graph of nectar rewards collected by colonies of a-bees versus the percentage of grid cells occupied by distractor flowers. Target flower density was fixed at 10%. Data points are the mean of 20 simulations; error bars  $\pm 1$  std. dev.

## Discussion, Conclusions and Future Work

In the current study we tested for the relative success of a serial scan as occurs in honeybees, and a parallel scan typical of bumblebees, considering previous evidence that honeybees evolved to forage in more tropical environments and bumblebees in temperate environments (Clark 1994; Dornhaus and Chittka 2004; Heinrich 2004; Dyer, Spaethe *et al.* 2008). This lead us to hypothesise that honeybees' serial scan may be more effective in environments where targets were not interspersed among distractors, and that bumblebees' parallel scan would be relatively more effective for foraging in heterogeneous environments where distractors and targets were intermingled. These hypotheses were supported by our simulation results as follows.

### Target only experiments

The ratio of nectar collected by the parallel a-bees to that collected by the serial a-bees in an environment without distractors approaches  $11,336:14,163 = 4:5$ . The serial mechanism is increasingly superior as target density increases.

This ratio can be derived analytically in the absence of distractors and a neighbourhood increasingly saturated with targets. For each traversal of the procedure in Fig. 2 a parallel a-bee requires  $ParallelTime + MoveTime + VisitTime$  to collect a reward unit. With our parameters (Tab. 1) this amounts to  $3+1+1 = 5$  simulation time steps. A serial a-bee's traversal of the procedure in Fig. 3 requires  $SerialTime + MoveTime + VisitTime = 2+1+1 = 4$  simulation time steps to collect a reward unit. Hence, a serial a-bee takes  $4/5^{th}$  the time of a parallel a-bee to collect a reward unit in this scenario.

We could have set our simulation parameters so that the a-bees' decision time wasn't the most time-consuming activity in its foraging cycle. This would potentially be more biologically plausible given that decision times as measured by Morawetz and Spaethe (2012) were generally less than 1.5 seconds (bumblebees) and 1.2 seconds (honeybees), and that the sum of travel and flower handling times may exceed these values in the real world by many seconds (Chittka, Gumbert *et al.* 1997; Chittka 2002). However, these time handling factors may also be complicated by interactions with other foraging insects in complex and competitive environments where being faster may allow the collection of rewards before competitors diminish resources (Burns and Dyer 2008). The impact of travel and flower handling times for real honeybees and bumblebees may impact on their relative success in different environments since each species may conceivably have differences in flight speed and flower handling skill under various real world conditions. But the scope of our experiments was only to test differences between the scanning mechanisms these species have been shown to employ. Hence, we neutralise flower handling and travel time differences by assigning equal values to *VisitTime* and *MoveTime* parameters for the two a-bee species.

It was found that real bumblebees reduce their decision time slightly, by a statistically significant amount, as the number of visible targets increases from one to four (Morawetz and Spaethe 2012). Hence, the gap we report between parallel and serial mechanisms may be decreased slightly from that shown for our a-bees as target density increases (Fig. 5).

### Target and distractor experiments

A distractor density of only 1% is sufficient to level the effectiveness of the serial and parallel scanning mechanisms. Any increase beyond this improves the parallel scanning mechanism's superiority over the serial scan. This is interesting as it implies that in many biologically plausible scenarios, parallel search would be the more efficient approach for collecting nutrition from rewarding flowers whilst avoiding dissimilar colored distractors. Why then mightn't honeybees have evolved a more efficient visual scan? A potential answer is provided in figure 5 where, in an environment with no distractors (as would be present in many tropical scenarios where large trees flower) fast serial search is more effective. Indeed, in studies that have compared the efficiency of honeybees at collecting nutrition in temperate and tropical environments, and where the capacity of the

honeybees to recruit nest mates to share in resource gathering was experimentally manipulated, research has found honeybees to be most effective in tropical environments (Dornhaus and Chittka 2004). Thus our current findings that there are biologically plausible conditions that best suit either a parallel or serial search mechanism, adds weight to the possibility of environmental conditions leading to the evolution of different visual capabilities in different bee species (Dyer, Spaethe *et al.* 2008; Morawetz and Spaethe 2012).

As noted above, the biased random walk has no impact on the relative success of the scanning mechanisms. However in absolute terms, in our simulations, the turning bias we implemented improved the parallel a-bees' foraging success by 20.48% (std. dev. 0.85) and the serial a-bees' success by 24.7% (std. dev. 2.4). This certainly leaves it open for those interested in optimal foraging to simulate more biologically plausible navigation schemes, however this was outside the scope of our study.

### Future work

Our findings suggest it would be valuable to compare tropical and temperate floral distributions and how these may have affected the processing capabilities of different bee species. This work is potentially useful because it has been suggested that climate change may lead to either spatial or temporal mismatches in the availability of flower resources and pollinators (Hegland, Nielsen *et al.* 2009), and indeed such mismatches could be influenced by the visual capacities of bees. Currently this is unknown, and testing bee species of high pollination value with flowers of different patchiness may provide important insights for future resource management.

### Acknowledgements

This research was supported under Australian Research Council's *Discovery Projects* funding scheme (project numbers DP0878968, DP0987989, DP130100015).

### References

- Axelrod, R. (2003). "Advancing the art of simulation in the social sciences." *Japanese Journal for Management Information Systems - Special Issue on Agent-Based Modeling*: 19.
- Burns, J. G. (2005). "Impulsive bees forage better: the advantage of quick, sometimes inaccurate foraging decisions." *Animal Behaviour* **70**: e1-e5.
- Burns, J. G. and A. G. Dyer (2008). "Diversity of speed accuracy strategies benefits social insects." *Current Biology* **18**: R953–R954.
- Chittka, L. (2002). "The influence of intermittent rewards on learning to handle flowers in bumblebees." *Entomologia generalis* **26**(2): 85-91.
- Chittka, L., A. Gumbert and J. Kunze (1997). "Foraging dynamics of bumble bees: correlates of movements within and between plant species." *Behavioural Ecology* **8**: 239-249.



- Chittka, L., J. D. Thomson and N. M. Waser (1999). "Flower constancy, insect psychology, and plant evolution." *Naturwiss* **86**: 361-377.
- Clark, D. A. (1994). Plant demography. in *La Selva—ecology and natural history of a neotropical rain forest*. L. A. McDade, K. S. Bawa, H. A. Hespenheide and G. S. Hartshorn. Chicago, University of Chicago Press: 90-105.
- Dafni, A. (1984). "Mimicry and deception in pollination." *Annual Review of Ecology and Systematics* **15**: 259-278.
- DeAngelis, D. L. and W. M. Mooij (2005). "Individual-based modelling of ecological and evolutionary processes." *Annu. Rev. Ecol. Evol. Syst.* **36**: 147-168.
- Dorin, A., K. B. Korb and V. Grimm (2008). Artificial-Life Ecosystems: What are they and what could they become? *Eleventh International Conference on Artificial Life*. S. Bullock, J. Noble, R. A. Watson and M. A. Bedau, MIT Press: 173-180.
- Dornhaus, A. and L. Chittka (1999). "Evolutionary origins of bee dances." *Nature* **401**: 38.
- Dornhaus, A. and L. Chittka (2004). "Why do honeybees dance?" *Behavioural Ecology and Sociobiology* **55**: 395-401.
- Dornhaus, A., F. Klügl, C. Oechslein, F. Puppe and L. Chittka (2006). "Benefits of recruitment in honey bees: effects of ecology and colony size in an individual-based model." *Behavioural Ecology* **17**(336-344).
- Duchateau, M. J. and H. H. W. Velthuis (1988). "Development and reproductive strategies in *Bombus terrestris* colonies." *Behaviour* **107**: 186-207.
- Dyer, A. G., S. Boyd-Gerny, S. McLoughlin, M. G. P. Rosa, V. Simonov and B. B. M. Wong (2012). "Parallel evolution of angiosperm colour signals: common evolutionary pressures linked to hymenopteran vision." *Proceedings of the Royal Society of London B* **279**: 3605-3615.
- Dyer, A. G., A. Dorin, V. Reinhardt and M. G. P. Rosa (2012). "Colour reverse learning and animal personalities: the advantage of behavioural diversity assessed with agent-based simulations." *Nature Precedings*(March): 20.
- Dyer, A. G., A. C. Paulk and D. H. Reser (2011). "Colour processing in complex environments: insights from the visual system of bees." *Proceedings of the Royal Society B* **278**: 952-959.
- Dyer, A. G., J. Spaethe and S. Prack (2008). "Comparative psychophysics of bumblebee and honeybee colour discrimination and object detection." *Journal of Computational Physiology A* **194**: 614-627.
- Frisch, K. v. (1967). "The Dance Language and Orientation of Bees". Cambridge, USA, Harvard University Press.
- Giurfa, M. (2004). "Conditioning procedure and color discrimination in the honeybee *Apis mellifera*." *Naturwissenschaften* **91**: 228-231.
- Giurfa, M., J. Núñez and W. Backhaus (1994). "Odour and colour information in the foraging choice behaviour of the honeybee." *Journal of Comparative Physiology A* **175**(6): 773-779.
- Grimm, V. (1999). "Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future?" *Ecological Modelling* **115**: 129-148.
- Grimm, V. and S. F. Railsback (2005). "Individual-based Modeling and Ecology", Princeton University Press.
- Grimm, V., E. Revilla, U. Berger, F. Jeltsch, W. M. Mooij, S. F. Railsback, H.-H. Thulke, J. Weiner, T. Wiegand and D. L. DeAngelis (2005). "Pattern-Oriented Modelling of Agent-Based Complex Systems: Lessons from Ecology." *Science* **310**: 987-991.
- Hegland, S. J., A. Nielsen, A. Lázaro, A. L. Bjerknes and Ø. Totland (2009). "How does climate warming affect plant-pollinator interactions?" *Ecological Letters* **12**: 184-195.
- Heinrich, B. (2004). "Bumblebee economics". Cambridge, Harvard University Press.
- Hogeweg, P. and B. Hesper (1983). "The Ontogeny of the Interaction Structure in Bumble Bee Colonies: A MIRROR Model." *Behavioural Ecology and Sociobiology* **12**: 271-283.
- Huston, M., D. DeAngelis and W. Post (1988). "New Computer Models Unify Ecological Theory." *BioScience* **38**(10): 682-691.
- Judson, O. P. (1994). "The rise of the individual-based model in ecology." *Trends in Ecology and Evolution* **9**(1): 9-14.
- Kjøl, M., A. Nielsen and N. C. Stenseth (2011). "Potential Effects of Climate Change on Crop Pollination". Rome, Food and Agricultural Organization.
- Lythgoe, J. N. (1979). "The ecology and vision". Oxford, U.K., Clarendon Press.
- Morawetz, L. and J. Spaethe (2012). "Visual attention in a complex search task differs between honeybees and bumblebees." *Journal of Experimental Biology* **215**: 2515-2523.
- Pyke, G. H. (1984). "Optimal Foraging Theory: A Critical Review." *Annual Review of Ecology and Systematics* **15**: 523-575.
- Srinivasan, M. V. (2011). "Honeybees as a model for the study of visually guided flight, navigation, and biologically inspired robotics." *Physiological Reviews* **91**(2): 413-460.
- Streinzer, M., H. F. Paulus and J. Spaethe (2009). "Floral colour signal increases short-range detectability of a sexually deceptive orchid to its bee pollinator." *Journal of Experimental Biology* **212**: 1365-1370.
- Treisman, A. M. and G. Gelade (1980). "A feature-integration theory of attention." *Cognitive Psychology* **12**: 97-136.
- Treisman, A. M. and S. Gormican (1988). "Feature analysis in early vision: evidence from search asymmetries." *Psychological Review* **95**: 15-48.
- VanRullen, R., T. Carlson and P. Cavanagh (2007). "The blinking spotlight of attention." *Proceedings of National Academy of Science USA* **104**: 19204-19209.
- Vries, H. d. and J. C. Biesmeijer (1998). "Modelling collective foraging by means of individual behaviour rules in honey-bees." *Behavioural Ecology and Sociobiology* **44**(2): 109-124.
- Waddington, K. D. (1980). "Flight Patterns of Foraging Bees Relative to Density of Artificial Flowers and Distribution of Nectar." *Oecologia* **44**(2): 199-204.
- Wertlen, A. M., C. Niggebrugge, M. Vorobyev and N. H. d. Ibarra (2008). "Detection of patches of coloured discs by bees." *Journal of Experimental Biology* **211**: 2101-2104.

# Using MapReduce Streaming for Distributed Life Simulation on the Cloud

Atanas Radenski

Chapman University, Orange, California  
Radenski@chapman.edu

## Abstract

Distributed software simulations are indispensable in the study of large-scale life models but often require the use of technically complex lower-level distributed computing frameworks, such as MPI. We propose to overcome the complexity challenge by applying the emerging MapReduce (MR) model to distributed life simulations and by running such simulations on the cloud. Technically, we design optimized MR streaming algorithms for discrete and continuous versions of Conway's *life* according to a general MR streaming pattern. We chose *life* because it is simple enough as a testbed for MR's applicability to a-life simulations and general enough to make our results applicable to various lattice-based a-life models. We implement and empirically evaluate our algorithms' performance on Amazon's Elastic MR cloud. Our experiments demonstrate that a single MR optimization technique called strip partitioning can reduce the execution time of continuous *life* simulations by 64%. To the best of our knowledge, we are the first to propose and evaluate MR streaming algorithms for lattice-based simulations. Our algorithms can serve as prototypes in the development of novel MR simulation algorithms for large-scale lattice-based a-life models<sup>1</sup>.

## Introduction

A-life has long relied on software simulations of the behavioral characteristics of living systems to facilitate the discovery of natural laws. Living systems involve vast numbers of evolving objects, and their software simulations can be data and computationally intensive. Large-scale life models may not fit in the memory of an average workstation, a challenge that can be overcome with the development of distributed a-life software. Distributed scientific simulations are usually implemented by using low-level libraries, such as MPI, which are difficult to program and require the development of custom fault-tolerance and load-balancing schemes --- a major challenge for scientists. Compute clusters to run distributed simulations can be expensive to build and complex to maintain. We believe that the complexity challenges of distributed life simulations can be overcome by applying the emerging higher-level MapReduce (MR) model to life simulations and by running such simulations on the cloud.

MR was initially developed to specifically satisfy Google's needs for large-scale distributed processing of unstructured

text data [Dean and Ghemawat, 2008]. The subsequent implementation of MR within Apache's open-source Hadoop framework [White, 2012] stimulated the development of wide range of MR applications in diverse areas, including sets and graphs; AI, machine learning and data mining; bioinformatics; image and video; evolutionary computing; and statistics and numerical mathematics [Radenski and Norris, 2013]. MR users develop serial code that is automatically executed in parallel by the MR engine in a fault-tolerant and load-balanced manner. The simplicity of the MR model and the built-in fault tolerance and load-balancing functionality of the MR engine can be beneficial in the development of data-intensive distributed scientific applications in general and life software simulations in particular.

Our general goal in this paper is to investigate the applicability of the MR model to large-scale distributed a-life simulations. To do so, we focus on life models that are based on cellular automata (CA) because CA are relatively easy to parallelize and have been used in life modeling and simulation from the early days of a-life. Indeed, a-life and cellular automata share a closely tied history which began with the CA works of John von Neumann [Von Neumann and Burks, 1966] and continued with the development and exploration of the game of life --- or simply *life* --- by John Conway [Gardner, 1970; Bays, 2010]. Conway's *life* is a 2D CA with two states (alive/dead). A popular *life* variation adopts a nondiscrete representation with states that are continuously valued between zero and one [Peper et al., 2010]. We use the terms *discrete life* and *continuous life* correspondingly to distinguish between the two *life* models.

CA has been recognized as historically the most fundamental paradigm of a-life [Conti, 2008], and *life* has become known as the prototypical CA [Hoekstra et al., 2010]. Hence, we consider *life* to be a most suitable candidate for this first study of the applicability of the MR model to large-scale distributed life simulations. Technically, we develop and empirically evaluate MR *life* algorithms for the discrete and continuous *life* models. We also discuss how our algorithms can be transformed into algorithms for other versions of *life*, such as 3D *life* and *life* with larger neighborhoods. We outline a general MR streaming pattern that encapsulates the common general features of the *D-Life* and *C-Life* algorithms and can be followed for the design of other lattice-based simulation algorithms in the MR streaming model. Our algorithms are by no means limited to *life* and can be used as prototypes for developing large-scale MR simulation algorithms for other CA-based life models.

<sup>1</sup> This work was performed by Atanas Radenski as a guest faculty at Argonne National Laboratory in Illinois while on a sabbatical leave from Chapman University, California in spring 2013.

Cloud computing is the use of hardware and software as service --- remotely, on-demand, and on a pay-per-use basis. Cloud computing can be viewed as a descendant of the well-established grid computing, enhanced with instant provisioning and utility computing. Elastic MR is one of the services provided by the Amazon's cloud computing platform, Amazon Web Services (AWS). Elastic MR is in fact Apache Hadoop hosted on AWS's Elastic Compute Cloud. We develop MR *life* algorithms in the Hadoop version of the MR model and then host the algorithms' execution on Elastic MR for the purpose of empirical performance evaluation. With AWS, we can launch various Elastic MR clusters as they are needed for our cost-effective experiments. We choose AWS because it is the first and currently the largest publicly available cloud computing platform.

Previously unknown complex self-organizing behavior of life models can be discovered by simulating vast numbers of generations for very large-scale model configurations. Such large-scale simulations may not fit on individual workstations but can be conveniently implemented in MR and executed on the cloud in a cost-effective, pay-per-use fashion. Our MR *life* algorithms and their empirical performance evaluation are intended to enhance scientists' understanding of the potential of MR and cloud computing in a-life research and open new opportunities for distributed large-scale life simulations.

The rest of this paper contains three sections. The first section describes how *life* simulations can be represented in the MR model. The second section discusses related work. The third section presents conclusions and possibilities for future work.

## Placing Life on MapReduce

This section introduces selected features of the MR model, defines MR algorithms for discrete *life* and for continuous *life*, offers a general MR streaming pattern, and ends with an empirical evaluation of the MR algorithms on the cloud.

### MapReduce Models and Frameworks

**Standard MR model.** In the *standard MR model*, user-defined serial *Map* and *Reduce* methods transform in parallel an input set of key-value (KV) pairs into an output set of KV-pairs. Initially, *Map* is applied in parallel to individual KV-pairs from the input set to produce a first intermediate set of KV-pairs.

$$Inter1 = \{(k2, v2)\} = \{Map(k1, v1) \mid (k1, v1) \in Input\}$$

This set of KV-pairs is then automatically transformed by MR into a second intermediate set of KV-pairs in which all intermediate pairs with the same key are sorted and grouped together, creating a single KV-pair for each intermediate key.

$$Inter2 = \{(k2, list(v2))\} = MR-Sort-and-Group(Inter1)$$

*Reduce* then is applied in parallel to individual KV-pairs from the second intermediate set to produce an output set of KV-pairs.

$$Output = \{(k3, v3)\} = \{Reduce(k2, list(v2)) \mid (k2, list(v2)) \in Inter2\}$$

Input, intermediate, and output keys and values may or may not belong to different domains.

Consider for example the problem of counting word frequencies in a text document. In standard MR word count, input KV pairs represent document's lines (Table 1). The standard MR engine parses the input keys and values and provides them as ready-to-use KV arguments to *Map* and *Reduce*. *Map* method invocations (Figure 1) parse individual lines (automatically submitted to *Map* through the *value* parameter) and produce the first intermediate set of KV pairs, where individual words serve as keys and 1s serve as values (Table 1). MR then automatically sorts and groups the first intermediate set into a second intermediate set (Table 1). Finally, *Reduce* method invocations (Figure 1) sumup grouped values to output the final count for each word (Table 1). Note that this particular *Map* method ignores all input keys. For practical convenience, MR frameworks automatically generate some default keys for existing text documents.

Table 1: Standard MR – data sample

Input	Inter1	Inter2	Output
1 to be or	to 1	be 1 1	be 2
2 not to be	be 1	not 1	not 1
	or 1	or 1	or 1
	not 1	to 1 1	to 2
	to 1		
	be 1		

```

1: class Mapper:
2:   method Map (key, value):
3:     for word ∈ value:
4:       Emit (key=word, value =1);
1: class Reducer:
2:   method Reduce (key, list-of-values):
3:     sum = 0;
4:     for value ∈ list-of-values:
5:       sum +=value;
6:     Emit (key, value = sum);

```

Figure 1: Word count in the standard MR model

**MR frameworks.** The MR model has been implemented in three principal types of frameworks: distributed MR (targeted at clusters of workstations) [Dean and Ghemawat, 2008], shared-memory MR (targeted at multicore, multiprocessors workstations) [Talbot et al., 2011], and GPU MR [He et al., 2008]. Distributed MR frameworks are the most popular in practical computing. Any distributed MR framework incorporates an MR engine and a distributed file system (DFS) to hold the input, intermediate, and output datasets.

Google was the first to develop a proprietary distributed MR framework, which has been available and used only internally [Dean and Ghemawat, 2008]. The popularity of the MR model grew significantly with the release by Apache of the open-source Hadoop framework [White, 2012], which extended the standard MR model with additional functionality, such as *MR streaming*. Hadoop is now the defacto standard MR framework, and we therefore target our distributed MR *life* algorithms to Hadoop (see [Lee et al., 2012] for advantages and pitfalls of Hadoop MR). For the rest

of this paper we omit “Hadoop” in references to the Hadoop MR.

The MR engine invokes *Map* and *Reduce* methods within persistent tasks that are distributed over the Hadoop cluster; we refer to such persistent tasks as *mappers* and *reducers*. The MR engine uses intermediate keys to partition all intermediate KV-pairs among available reducers and to sort all KV-pairs that are fed into the same reducer. Hence, all intermediate KV-pairs with the same key are submitted to the same reducer in sorted order, although the same reducer can be assigned to handle several different keys [Radenski, 2012].

The MR framework is implemented in Java, and standard MR algorithms target the MR Java API, thus requiring significant Java expertise. In contrast to standard MR, MR streaming algorithms target higher-level languages such as Python. Because MR streaming is easier to understand and modify by domain scientists, who may not be Java experts but can work well with Python, we chose to develop MR streaming algorithms for distributed *life* simulations. Such MR streaming algorithms can be straightforwardly implemented in Python. If needed, MR streaming algorithms can be transformed into equivalent standard MR algorithms.

**MR streaming model.** In an essential departure from the standard MR semantics, MR streaming sorts but does not group intermediate same-key KV-pairs at all, and the *Reduce* method must handle multiple occurrences of the same key with corresponding partial values (rather than a single list of grouped values as in standard MR). The semantics of the *MR streaming model* can be specified as follows.

$$\begin{aligned} Inter1 &= \{(k2, v2)\} = \cup \{Map(\{(k1, v1)\}) \mid \{(k1, v1)\} \subseteq Input\} \\ Inter2 &= \{(k2, v2)\} = MR\text{-}Sort(Inter1) \\ Output &= \{(k3, v3)\} = \cup \{Reduce(\{(k2, v2)\}) \mid \{(k2, v2)\} \subseteq Inter2\} \end{aligned}$$

While the standard MR engine parses data sets’ keys and values and provides them as ready-to-use KV arguments to *Map* and *Reduce*, the streaming MR engine provides all data via the standard input stream. Consequently, *Map* and *Reduce* must parse the input into keys and values on their own. Parsing input in a higher-level language is not necessarily a disadvantage of MR streaming: such parsing can be more straightforward and flexible than using the relatively complicated and rigid Java API to manage the input format in standard MR [Radenski and Norris, 2013].

In MR streaming word count, document’s lines are viewed as input KV pairs represent with empty keys (Table 2). The *Map* method (Figure 2) parses individual lines and produces an intermediate set of KV pairs in which individual words serve as keys (Table 2). MR then automatically sorts the first intermediate set into a second intermediate set (Table 2). Finally, the *Reduce* method (Figure 2) uses a loop to sumup sorted values then outputs the final count for each word (Table 2).

The distributed execution of mappers and reducers forms a single MR streaming job (Figure 3). Several MR jobs can be iterated so that the output from one job is used as the input for the next one. All input datasets, intermediate datasets, and output datasets for MR jobs are stored in the DFS. Iterative MR processing can be achieved by using functionality that is

external to the MR model or by using non-standard extensions of the MR model itself.

Table 2: MR streaming – data sample

Input	Inter1	Inter2	Output
to be or not to be	to 1 be 1 or 1 not 1 to 1 be 1	be 1 be 1 not 1 or 1 to 1 to 1	be 2 not 1 or 1 to 2

```

1: class Mapper:
2:     method Map():
3:         for line ∈ stdin:
4:             for word ∈ line:
5:                 Emit(key=word, value = 1);
1: class Reducer:
2:     method Reduce():
3:         last-word = None;
4:         for line ∈ stdin:
5:             current-word, value = Parse(line);
6:             if current-word ≠ last-word:
7:                 if last-word ≠ None:
8:                     Emit(last-word, sum)
9:                 last-word = current-word; sum = 0
10:            sum += value;
11:            Emit(last-word, sum)

```

Figure 2: Word count in the MR streaming model

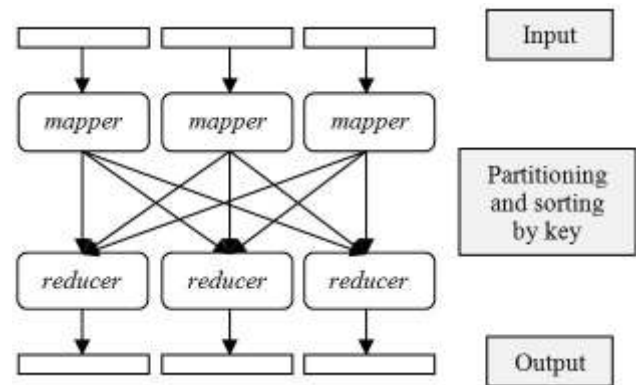


Figure 3: MR streaming job dataflow

### Discrete Life in MapReduce

*Discrete life* is a CA with two states (dead/alive) over an infinite 2D lattice that evolves according the 2,3/2 rule: An alive cell with 2 or 3 alive neighbors in its Moore neighborhood stays alive, and a dead cell with 2 alive neighbors becomes alive. (As defined earlier, *discrete life* is our term for Conway’s *life*, as opposed to continuous life.) This rule can be generalized as  $E_1, E_2, \dots / F_1, F_2, \dots$  to define a family of *life-like* CA [Bays 2010]. The evolution of *discrete life* is deterministic and is completely defined by its initial state.



Standard MR and MR streaming both operate on KV-pairs. In MR streaming, each KV-pair has to be represented as a single line of text. Input, output, and intermediate datasets in 2D *discrete life* simulations represent living cells as KV-pairs in which the key part consists of the cells' coordinates (*row*, *col*) and the value part is empty. Cells that are not included in the dataset are assumed dead (Table 3).

Table 3: *Life* data representation – data sample

<i>Discrete life</i>	<i>Continuous life</i>
	1 1 0.0
	1 2 0.5
	1 3 0.0
2 1	2 1 1.0
2 2	2 2 1.0
2 3	2 3 1.0
	3 1 0.0
	3 2 0.5
	3 3 0.0

The initial *life* state is stored in one or more text files on the DFS before processing. An MR streaming job automatically splits the input dataset into independent blocks, which are submitted to mappers, one line per cell at a time (Figure 3). Mappers process individual cells and emit intermediate KV-pairs that are partitioned and sorted by the MR engine and then input into reducers. Reducers process intermediate KV-pairs and emit output results that are stored back on the DFS, one output file per reducer. The output dataset can then be used as the input for a subsequent MR streaming job. Hence, a single *life* simulation step is implemented as a single MR streaming job. A multi-step *life* simulation can be realized as a MR streaming job iteration by using tools that are outside of the MR streaming model. This pattern is followed for both *discrete life* simulations (discussed in this subsection) and *continuous life* simulations (discussed in the next subsection).

A single *discrete life* simulation step can be implemented by means of a technique known as MR *message passing* [Lin and Schatz, 2010]. For each input cell (by definition alive) a mapper can emit intermediate KV-pairs that are interpreted as messages to all of the cell's neighbors, alive and dead. Such messages from a living cell simply notify all of the cell's neighbors --- living or alive, and including the cell itself --- that the living cell belongs to those cells' neighborhoods. Messages to the same cell are dispatched to the same reducer, and each reducer receives all its messages in sorted order. This enables a reducer to count the living neighbors of individual cells, to determine the cells' next states (dead/alive) and to emit only living cells.

MR message passing can generate numerous small messages. Higher message volume can become a MR network bottleneck and be detrimental to performance. The number of messages can be reduced by applying local in-mapper aggregation (LA) optimizations [Lin and Dyer, 2010]. LA is applied to *discrete life* simulation in MR streaming as follows. For each neighbor of each input cell, the mapper increments (rather than emit immediately) a cell's counter within a local in-memory hash. Aggregated counts for each cell are emitted just before the mapper termination. Such aggregated counts for the same cell are summed up by a single reducer that determines the next state of the cell. A *discrete life* single-step

simulation algorithm in MR streaming referred to as *D-Life*, is shown in Figure 4. This algorithm assumes no size limits on the lattice and can operate on very large *discrete life* instances.

```

1: class Mapper:
2:   method Map ():
3:     hash =  $\emptyset$ 
4:     for line  $\in$  stdin:
5:       cell = (row, col) = Parse (line)
6:       Emit (cell, tag = Alive, count = None)
7:       for neighbor in Neighborhood (cell):
8:         hash[neighbor] += 1
9:       for cell in hash:
10:        Emit (cell, tag = None, count = hash[cell])
11: class Reducer:
12:   method Reduce ():
13:     last-cell = None;
14:     for line  $\in$  stdin:
15:       current-cell, tag, count = Parse (line);
16:       if current-cell  $\neq$  last-cell :
17:         if last-cell  $\neq$  None:
18:           if Next-State-Is-Alive (last-cell):
19:             Emit (last-cell)
20:           last-cell = current-cell; alive-neighbors = 0
21:           alive-neighbors += count;
22:       if Next-State-Is-Alive (last-cell): Emit (last-cell)

```

Figure 4: Single-step *discrete life* simulation algorithm in MR streaming (*D-Life*)

In the *D-Life* algorithm (Figure 4), the 2,3/2 rule and the Moore neighborhood of Conway's *discrete life* are encapsulated in methods Next-State-Is-Alive and Neighborhood. Hence, the *D-Life* algorithm can be applied to simulate other *life*-like CA by merely adapting the two methods to alternative rules and neighborhoods. *Larger than life* (LtL), for example, is a family of CA generalizing *discrete life* to large neighborhoods and general birth and survival thresholds [Evans, 2010]. Any LtL instance can be simulated by the *D-Life* algorithm with methods Next-State-Is-Alive and Neighborhood defined to properly implement the specific rule and neighborhood. Variations of *discrete life* in larger dimensions [Bays, 1987] can be simulated with the *D-Life* algorithm by merely extending the cell representation to a larger number of dimensions.

## Continuous Life in MapReduce

*Continuous life* is a CA with continuously valued states from the 0..1 range. *Continuous life* employs a transition rule that is formulated as a nonlinear expression with a temperature parameter  $T$  and two parameters:  $E0$  (energy shift parameter) and  $x0$  (state shift parameter). In the limit  $T \rightarrow 0$  and with appropriately chosen values for  $E0$  and  $x0$ , the behavior of *continuous life* coincides with that of *discrete life* [Adachi et al., 2004].

We rewrote the original *continuous life* transition function [Adachi et al., 2004] in an equivalent form that is more suitable for a MR implementation. Let us denote the state of a given cell at time  $t$  as  $S(\text{cell}, t)$ . In *continuous life*, each cell

undergoes transitions at discrete time steps according to the following set of equations.

- (1)  $S(\text{cell}, t + 1) = F(E(H(\text{cell}, t)))$
- (2)  $F(z) = b / (b + 1)$  where  $b = \exp(2 * z / T)$
- (3)  $E(x) = E0 - (x - x0)^2$
- (4)  $H(\text{cell}, t) = S(\text{cell}, t) + 2 * \sum_{\text{cell}' \in \text{Neighborhood}(\text{cell})} S(\text{cell}', t)$

For *continuous life* simulations in the MR streaming model, input, output, and intermediate datasets represent living cells as KV-pairs in which the key part consists of the cells' coordinates (*row*, *col*) and the value part is the cell's *state*. Hence, all cells are explicitly represented in *continuous life*'s datasets, in contrast to *discrete life*'s datasets which contain living cells alone (Table 3).

To implement a single *continuous life* simulation step in the MR streaming model, we combine the MR message passing technique with the *strip partitioning* optimization technique. MR message passing was introduced in the previous subsection for simulation of *discrete life*. Strip partitioning was proposed originally for large-scale relaxation algorithms [Radenski and Norris, 2013].

A mapper inputs KV-pairs of the form (*cell*, *state*) from the DFS. For each neighbor *cell'* of the input *cell*, the mapper aggregates in memory partial values of the term  $H(\text{cell}', t)$  according to equation (4). Locally aggregated partial values for each *cell'* are emitted just before the mapper termination as intermediate KV-pairs. Such aggregated values for the same *cell'* are summed up by a single reducer to determine the complete value of  $H(\text{cell}', t)$ . The reducer then uses this complete value to calculate the next state of the *cell'* according to equations (1) --- (3) and to emit the new state to the DFS.

In MR, the intermediate output of each mapper is directed to specific reducers by the MR *partitioner*. In general, given an intermediate key-value record, the default MR partitioner hashes the key into a reducer index for the record, balancing the load among reducers [Radenski and Norris, 2013]. In *continuous life* simulation, intermediate records are in the form (*key*= *cell'*,  $H_{\text{partial}}(\text{cell}', t)$ ). Because of hashing, adjacent cells are likely to be directed to different reducers, and thus output into different DFS files (because each reducer's output is placed by the MR engine into a separate DFS file). At the next simulation step, adjacent cells are likely to be submitted to different mappers (because such adjacent cells are likely to have been output to different DFS files at the previous simulation step and because different input files are submitted to different mappers). Thus, the default MR partitioner tends to disperse neighborhoods and reduce data locality, which impairs local in-mapper aggregation and is detrimental for performance. [Radenski and Norris, 2013] Fortunately, the strip partitioning optimization technique can help reduce dispersion and preserve data locality. With strip partitioning, a mapper sends whole strips of consecutive CA lattice rows to the same reducer. Technically, the mapper outputs intermediate KV-pairs of the form:

$$(\text{key}=(\text{strip}, \text{cell}'), \text{value}=H_{\text{partial}}(\text{cell}', t)),$$

where *strip* is an index that identifies individual strips of CA lattice rows. The *strip* index is calculated as  $\text{strip} = \text{row} /$

*strip-length* where *strip-length* is a simulation parameter. Because the *strip* index is part of the mappers' intermediate output, all cells from the same strip will be directed to the same reducer during partitioning (see Figure 3), possibly from different mappers. Therefore, such adjacent cells will remain in the same output file as produced by the reducer. This strategy preserves data locality during iterative simulation and promotes performance. The *strip* index is emitted by mappers to facilitate partitioning alone and is ignored by reducers upon its receipt as part of the key.

A *continuous life* single-step simulation algorithm in MR streaming based on local in-mapper aggregation and strip partitioning (referred to as *C-Life*) is shown in Figure 5.

```

1: class Mapper:
2:   method Map ():
3:     hash = ∅
4:     for line ∈ stdin:
5:       cell, state = Parse (line)
6:       hash[cell] += state
7:       for neighbor in Neighborhood (cell):
8:         hash[neighbor] += 2*state
9:     for cell in hash:
10:      strip-number = cell.row / strip-length
11:      Emit (cell, strip-number, hash[cell])
1: class Reducer:
2:   method Reduce ():
3:     H = 0; last-cell = None
4:     for line ∈ stdin:
5:       strip-number, current-cell, in-value = Parse (line);
6:       if current-cell ≠ last-cell :
7:         if last-cell ≠ None:
8:           Emit (last-cell, state=F(E(H)))
9:           H = 0; last-cell = current-cell
10:      H += in_value
11:      Emit (last-cell, state=F(E(H)))

```

Figure 5: Single-step *continuous life* simulation algorithm in MR streaming (*C-Life*)

For practical purposes, we assume that *continuous life* evolves over a finite rectangular lattice with periodic boundary conditions. This assumption prevents the otherwise unlimited growth of datasets in iterative *continuous life* simulation. While the *continuous life*'s lattice is assumed finite, it can be potentially very large in an MR implementation.

In the *C-Life* algorithm (Figure 5), the *continuous life*'s transition rule and the neighborhood are encapsulated in separate methods that can be adapted to alternative rules and neighborhoods, without any changes to the *C-Life* algorithm proper.

## General MapReduce Streaming Pattern

We have designed the *D-Life* and *C-Life* algorithms by following a general MR streaming pattern. Our pattern is outlined in Figure 6. The pattern describes a family of MR streaming algorithms that execute as follows (Figure 6):

- *Map* processes and aggregates locally each input KV-pair. Processing may consist of various actions, such as

emitting the KV-pair (as done in *D-Life*) and/or performing mathematical operations (as done in both *D-Life* and *C-Life*). Aggregation involves storing intermediate results locally in a hash.

- Just before termination, *Map* emits all aggregated results as intermediate KV-pairs, with the optional use of strip partitioning (as done in *C-Life*).
- *Reduce* processes and accumulate locally each intermediate KV-pair. Partitioning and sorting by key (Figure 3) guarantee that all intermediate KV-pairs with the same key are submitted to the same reducer in an uninterrupted sequence. Processing may consist of various actions, such as performing mathematical operations (e.g., increment in both *D-Life* and *C-Life*). Accumulation involves storing intermediate processing data locally (such as *alive-neighbors* in *D-Life* and *H* in *C-Life*).
- *Reduce* ends the processing of each uninterrupted same-key KV-pair sequence by calculating the key's final value and emitting an output KV-pair accordingly. (In *D-Life*, this involves deciding whether a cell will be dead or alive; in *C-Life* this involves calculating the cell's next state.)

```

1: class Mapper:
2:   method Map ():
3:     for input-kv-pair ∈ stdin:
4:       Process-and-Aggregate ()
5:       Emit-All-Aggregated ()
1: class Reducer:
2:   method Reduce ():
3:     for intermediate-kv-pair ∈ stdin:
4:       if Current-Key-Is-Different-From-Previous-Key ():
5:         Emit (previous-key, Final-Value ())
6:         Initialize-Current-KV-Pair-Processing ()
7:         Process-and-Accumulate ()
8:         Emit (last-key, Final-Value ())

```

Figure 6: General MR streaming pattern

## Cloud Implementation and Empirical Evaluation

We implemented our *D-Life* and *C-Life* MR streaming algorithms in Python and then used the implementations for empirical algorithm evaluation on Amazon's Elastic MR cloud. We chose Python because it is higher-level language that significantly shortens development efforts and time in comparison with other mainstream languages, such as Java or C++. Our experiments were performed with Hadoop 1.0.3 on an Elastic MR cluster of up to 17 large instances, a master instance and up to 16 core instances.

We experimented with two versions of the *C-Life* algorithm, designated as *C-Life-16* and *C-Life-0*. The *C-Life-16* version uses a strip size of 16. Our preference to a strip size of 16 is based on some preliminary performance experiments with various strip sizes. The *C-Life-0* version does not use the strip partitioning optimization at all, hence its strip size of 0.

We ran *D-Life*, *C-Life-16*, and *C-Life-0* on the Elastic MR cloud to measure their execution times. Execution times for the first simulation step can be influenced by the initial data layout; but once data are shuffled by the first simulation step, execution times stabilize. We measured execution times for

the algorithms' second simulation steps over randomly generated square lattices. The initial datasets for *D-Life* was generated with alive cell probability of  $p=0.5$ . Recall that only alive cells are represented in *D-Life*'s datasets; in contrast, all cells and their states are explicitly represented in *C-Life* datasets. Given the same lattice size, *D-Life*'s datasets are smaller than *C-Life*'s datasets proportionally to  $p$ .

Table 4: *Same-data* performance (in min) on up to 16 nodes

Nodes	1	2	4	8	16
<i>D-Life</i>	16.3	22.6	11.6	6.9	4.2
<i>C-Life-0</i>	32.2	39.7	29.4	17.2	11.0
<i>C-Life-16</i>	37.4	19.9	9.5	6.8	4.0

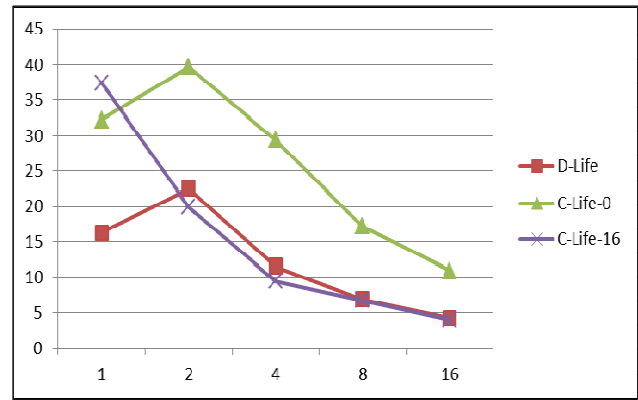


Figure 7: Visualization of *same-data* performance (in min, vertical axis) on up to 16 nodes (horizontal axis)

Table 5: *Same-load* performance (in min) on up to 16 nodes

Nodes	1	2	4	8	16
<i>D-Life</i>	2.0	3.1	3.2	4.0	4.2
<i>C-Life-0</i>	2.5	4.9	7.8	10.9	11.0
<i>C-Life-16</i>	2.9	3.6	3.3	3.8	4.0

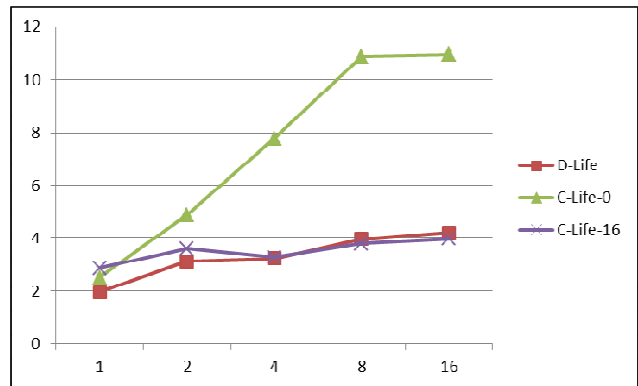


Figure 8: Visualization of *same-load* performance (in min, vertical axis) on up to 16 nodes (horizontal axis)

With all algorithms, we performed *same-data* and *same-load* performance evaluations. For *same-data* evaluation, we

measured algorithms' execution times in minutes on MR clusters with  $i$  core instances,  $i = 1, 2, 4, 8, 16$  over randomly generated square lattices of approximately  $16 \times 10^7$  cells (Table 4 and Figure 7). For same-load evaluation, we measured the algorithms' execution times in minutes on MR clusters with  $i$  core instances,  $i = 1, 2, 4, 8, 16$  over randomly generated square lattices of approximately  $i \times 10^7$  cells (Table 5 and Figure 8).

Our performance measurements demonstrate that strip partitioning optimization --- used in *C-Life-16* but neither in *C-Life-0* nor in *D-Life* --- gives a performance advantage of *C-Life-16* over *C-Life-0* and *D-Life*.

- The execution time of *C-Life-16* for a single simulation step on 16 task nodes is 64% less than the execution time of *C-Life-0* for the same task (see data in last columns of Table 4 or 5).
- The execution time of *C-Life-16* decreases in a smooth and predictable manner with the increase of task nodes, in contrast to both *D-Life* and *C-Life-0* (Figure 7).
- *C-Life-16* scales much better than *C-Life-0* and a little better than *D-Life* (Figure 8).
- In terms of absolute execution time, the performance of *D-Life* seems to rival that of *C-Life-16*, especially for larger numbers of task nodes (Figures 7 and 8); yet simulation of *discrete life* is computationally less intensive than simulation of *continuous life* and *D-Life* operates on smaller datasets than *C-Life* (given the same lattice size).

## Related Work

Cellular automata have been extensively studied since the early days of a-life [Langton, 1986]. A recent book offers a representative collection of approaches to the simulation of complex systems by CA [Hoekstra et al., 2010]. Another recent book covers current developments specifically in the game of Conway's *life* research [Adamatzky, 2010].

Our work on distributed MR *life* simulation builds on the MR strip-partitioning optimization originally introduced for a MR relaxation algorithm [Radenski and Norris, 2013]. Message passing was first studied in the context of data-intensive graph algorithms [Lin and Schatz, 2010] and later adapted to MR relaxation [Radenski and Norris, 2013]. Local in-mapper aggregation was originally designed to speed-up data-intensive text processing [Lin and Dyer, 2010] and was adapted to DNA sequence analysis [Radenski and Ehwerhemuepha, 2013] and relaxation [Radenski and Norris, 2013].

Our proposed *D-Life* and *C-Life* algorithms perform only a single *life* simulation step. A multistep *life* simulation is an iterative relaxation process that cannot be directly expressed in the pure MR parallelism model. We are among those who iterate pure MR steps by means of custom scripts expressed in common general-purpose languages. Others modify the pure MR model and implement new MR frameworks to facilitate iterative MR processing, such as iMapReduce [Zhang et al., 2012] and Twister [Ekanayake et al., 2010]. Potential ease of use and performance benefits of such iterative frameworks for multistep large-scale *life* simulations are yet to be studied.

Distributed MR relies exclusively on the DFS for the representation of intermediate datasets, including messages

passed by our *D-Life* and *C-Life* algorithms. Using the file system for message passing can be detrimental to performance but can be avoided with problems that fit entirely in memory. *In-memory MR* frameworks, such as Phoenix [Talbot et al., 2011] and M3R [Shinnar, 2012] aim to accelerate relatively small MR parallel applications by using hash tables to store intermediate key-value records in memory rather than on the DFS. Substantial speed-up benefits of in-memory frameworks for *life* simulations seem likely but are yet to be investigated.

*Discrete life*, a simple CA capable of generating diverse complex behavior, has stimulated many to design basic and advanced algorithms for its simulations and implement them in software. Basic serial and parallel implementations of *discrete life* have proven so worthy as to be incorporated in the computing curriculum [Wick, 2005; Hochstein et al., 2005]. Advanced *discrete life* simulation algorithms have been studied in traditional parallel computing models: shared memory [Ma et al., 2012], distributed [Xia et al., 2004], and mixed-mode [Smith and Bull, 2001]. To the best of our knowledge, we are the first to apply and evaluate the emerging MR model's applicability to distributed *life* simulations.

Various software frameworks have been developed and used to emulate of lattice-based a-life models since the early days of a-life, a trend that eventually began with the first *discrete life* programs. Lattice-based a-life software emulators continue to be used and developed [Komosinski and Adamatzky, 2010, Part II]. Notable examples include Discrete Dynamics Lab (DDLab), a set of tools for simulation of CA and other discrete structures [Wuensche, 2011]; NetLogo, a multiagent programmable modeling environment [Tisue and Wilensky, 2004]; and EINSTEIN, a multiagent simulator of land combat [Ilachinski, 2004]. We are the first to study the usability of MR in the a-life context.

## Conclusions and Future Work

In this paper, we investigate the applicability of the MR streaming model to the simulation of discrete and continuous *life* CA. We chose *life* CA because of their simplicity, a feature that makes them attractive as an initial test bed for distributed MR simulation approaches. We use MR message passing, local in-mapper aggregation, and strip partitioning to design the *D-Life* and *C-Life* algorithms for the simulation of discrete and continuous *life* correspondingly in the MR streaming model. We also formulate a general MR streaming pattern that we have followed in our design of *D-Life* and *C-Life* and that can be followed for the design of other CA simulation algorithms in the MR streaming model. We implement *D-Life* and *C-Life* on Amazon's Elastic MR cloud and empirically evaluate their performance. Our experimental results show that strip partitioning can reduce the execution time of continuous *life* simulations by 64%. To the best of our knowledge, we are the first to propose and evaluate MR streaming algorithms for lattice-based simulations.

In future projects, our proposed MR streaming algorithms can be used as prototypes in the development of novel MR simulation algorithms for large-scale CA in general and for lattice-based a-life models in particular. The field of applications of our approach can possibly be extended to the



field of multi agent simulation (MAS). MAS can be done in the standard MR model on a small Hadoop cluster [Sethia and Karlapalem. 2011] but the feasibility of MR streaming for larger scale MAS on the cloud is yet to be investigated.

Future work should aim at performance improvements. Performance improvements can be achieved by using standard MR instead of MR streaming and by using in-memory MR instead of distributed MR.

- The MR streaming engine does not aggregate intermediate KV-pairs at all, while the standard MR engine does it automatically; aggregation by the engine can be more efficient than custom aggregation in a higher-level language such as Python. For similar reasons, I/O, including KV-pair parsing, can also be more efficient in standard MR in comparison with MR streaming. With the use of standard MR instead of MR streaming, the tradeoffs is simplicity and ease of use for speed.
- Distributed MR frameworks use a DFS for all input, intermediate, and output datasets. The total I/O time can be much larger than the actual processing time. I/O performance losses can be offset by using in-memory MR frameworks, instead of distributed ones, for datasets that can fit in memory. With the use of in-memory MR instead of distributed MR, the ability to process unlimitedly large datasets is traded for speed.

As future work, our *D-Life* and *C-Life* MR streaming algorithms and our general MR streaming pattern can be translated into the standard MR model and ported onto an in-memory MR framework, to evaluate the performance gains with standard MR and in-memory MR.

**Acknowledgement.** Thanks are due to the anonymous reviewers for their valuable comments and recommendations.

## References

- Adachi, S., Peper, F., and Lee, J. (2004). The Game of life at finite temperature. *Physica D: Nonlinear Phenomena*, 198:182–196.
- Adamatzky, A., editor. (2010) *Game of Life Cellular Automata*. Springer, London, UK.
- Bays, C. (1987). Candidates for the Game of Life in three dimensions. *Complex Syst.* 1:373–400.
- Bays, C. (2010). Introduction to cellular automata and Conway's Game of Life. In [Adamatzky, 2010], pages 1–7.
- Conti, C. (2010). The enlightened Game of Life. In [Adamatzky, 2010], pages 453–464.
- Dean, J. and Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *CACM* 51:107–113.
- Ekanayake, J., Li, H., Zhang, B., Gunaratne, T., Bae, S.-H., Qiu, J., and Fox, G. (2010) Twister: A runtime for iterative MapReduce, In *19th ACM International Symposium on High Performance Distributed Computing*, pages 810–818, ACM, New York.
- Evans, K. (2010). Larger than Life's extremes: Rigorous results for simplified rules and speculation on the phase boundaries. In [Adamatzky, 2010], pages 179–221.
- Gardner, M. (1970). Mathematical games: The fantastic combinations of John Conway's new solitaire game "Life". *Sci. Am.* 223:120–123.
- He, B., Fang, W., Luo, Q., Govindaraju, N., and Wang, T. (2008). Mars: a MapReduce framework on graphics processors. In *17th International Conference on Parallel Architectures and Compilation Techniques*, pages 260–269. ACM, New York.
- Hochstein, L., Carver, J., Shull, F., Asgari, S., and Basili, V. (2005). Parallel programmer productivity: A case study of novice parallel programmers. In *Supercomputing 2005*, page 35. IEEE Computer Society, Los Alamitos, CA.
- Hoekstra, A., Kroc, J., and Sloot, P., editors. (2010). *Simulating Complex Systems by Cellular Automata*. Springer, Berlin.
- Ilachinski, A. (2004). *Artificial War: Multiagent-Based Simulation of Combat*. World Scientific Publishing Co., Inc., Hackensack, NJ.
- Komosinski, M. and Adamatzky, A. (2009). *Artificial Life Models in Software*, 2<sup>nd</sup> ed. Springer, Berlin.
- Langton, C. (1986). Studying artificial life with cellular automata. *Physica D: Nonlinear Phenomena*, 22:120–149.
- Lee, K.-H., Lee, Y.-J., Choi, H., Chung, Y. D., and Moon, B. (2012). Parallel data processing with MapReduce: a survey. *SIGMOD Rec.*, 40:11–20.
- Lin, J. and Dyer, C. (2010). *Data-Intensive Text Processing with MapReduce*. Morgan and Claypool, San Francisco, CA.
- Lin, J. and Schatz, M. (2010). Design patterns for efficient graph algorithms in MapReduce, In *8th Workshop on Mining and Learning with Graphs*, pages 78–85. ACM, New York.
- Ma, L., Chen, X., and Meng, Z. (2012). A Performance Analysis of the Game of Life Based on Parallel Algorithm, Technical Report, Computer Science and Engineering Department, Sichuan University.
- Peper, F., Adachi, S., and Lee, J. (2010). Variations on the Game of Life. In [Adamatzky, 2010], pages 235–255.
- Radenski, A. (2012). Distributed simulated annealing with MapReduce. *LNCS*, 7248:466–476.
- Radenski, A. and Ehwerhemuepha, L. (2013). Speeding-up codon analysis on the cloud with local MapReduce aggregation. *Information Sciences* (submitted).
- Radenski, A. and Norris, B. (2013) Distributed large-scale Laplace relaxation on the cloud with MapReduce. Technical Report, MCS Division, Argonne National laboratory, IL.
- Sethia, P. and Karlapalem, K. (2011) A multi-agent simulation framework on small Hadoop cluster. *Eng. Appl. Artif. Intell.* 24:1120–1127.
- Shinnar, A., Cunningham, D., Herta, B., and Saraswat, V. (2012) M3R: Increased performance for in-memory Hadoop jobs. *VLDB Endowment*, 5:1736–1747.
- Smith, L. and Bull, M. (2001). Development of mixed mode MPI/OpenMP applications. *Scientific Programming*, 9:83–98.
- Talbot, J., Yoo, R., and Kozyrakis, C. (2011). Phoenix++: Modular MapReduce for shared-memory systems. In *2nd International Workshop on MapReduce and Its Applications*, pages 9–16, ACM, New York.
- Tisue, S. and Wilensky, U. (2004). Netlogo: A simple environment for modeling complexity. In Minai A. and Bar-Yam Y., editors, *International Conference on Complex Systems*, pages 16–21. Westview Press, Cambridge, MA.
- Von Neumann, J. and Burks, A. (1966). *Theory of Self-Reproducing Automata*. University of Illinois, Urbana-Champaign.
- White, T. (2012). *Hadoop: The Definitive Guide*. O'Reilly, Sebastopol, CA.
- Wick, M.. (2005). Teaching design patterns in CS1: A closed laboratory sequence based on the game of life. *SIGCSE Bull.* 37:487–491.
- Wuensche, A. (2011). *Exploring Discrete Dynamics; The DDLab Manual*. Luniver Press, Frome, UK.
- Xia, H., Dail, H., Casanova, H., and Chien, A. (2004). The microgrid: Using online simulation to predict application performance in diverse grid network environments. In *Challenges of Large Applications in Distributed Environments*, pages 52–61. IEEE Computer Society, Los Alamitos, CA.
- Zhang, Y., Gao, Q., Gao, L., and Wang, C. (2012). iMapReduce: A distributed computing framework for iterative computation. *Journal of Grid Computing*, 10:47–68.

# Balancing the Costs and Benefits of Learning Ability

Kai Olav Ellefsen

Department of Computer and Information Science,  
Norwegian University of Science and Technology  
email: kaiolae@idi.ntnu.no

## Abstract

We study the costs and benefits of plasticity by evolving agents in environments with different rates of environmental change. Evolution allows both hard-coded strategies and learned strategies, with learning rates varying throughout life. We observe a range of change rates where the balance of costs and benefits are just right for evolving learning. Inside this range, we see two separate strategies evolve: lifelong plasticity and sensitive periods of plasticity. Sensitive periods of plasticity are found to reduce the learning cost while retaining the benefits of learning. This affects the evolutionary process, by limiting genetic assimilation of learned characteristics, making agents able to remain adaptive after relatively long periods of environmental stability.

## Introduction

Learning has been selected for by the process of natural evolution because it increases the fitness of individuals. This ability has both *advantages* and *costs* associated with it. The costs range from the energetic cost of maintaining a machinery for learning, to the cost having to do some potentially fatal trial and error to enable learning. The result of these costs is that evolution will attempt to minimize the amount of learning and, when possible, replace plasticity with innate strategies. Together, the costs and benefits of learning lead evolution to find a solution that learns just as much as is necessary and at the times in the lives of individuals when learning is most beneficial.

In this paper we study two phenomena related to plasticity regulation, to gain a better understanding of how they affect each other. The first one is the *Baldwin effect*, which describes plasticity regulation across *generations*. The second is *sensitive periods*, which describes plasticity regulation across individual lifetimes. An introduction to these phenomena, and related research on them is given in the next section.

## Background

### Learning- Costs and Benefits

The benefits of learning are frequently documented in studies of interactions between evolution and learning (see for

instance Floreano and Urzelai (2001), Littman (1995), and Nolfi et al. (1994)). When studying interactions between evolution and learning, it is important to also remember that learning has a *cost*. It is the balance between the cost and benefit of learning that decides the final learning strategies followed by individuals resulting from an evolutionary process. A comprehensive overview of the costs and benefits of learning is outside the scope of this paper. See Mayley (1996a) for a good overview of these factors. An implication of the cost of plasticity, is that plasticity in organisms needs to have *adaptive value*. When possible, natural selection should reduce costs by replacing plastic responses with genetic mechanisms.

Costs and benefits will vary significantly between individuals and even in single individuals in different situations. For instance the benefit of learning will be larger in an infant than an adult who has already learned the most important rules for gathering food and avoiding predators. See Turney (1996) for a comprehensive discussion of the trade-offs between plasticity and stability, and how this changes in different circumstances.

**Modeling the cost of plasticity** Kerr and Feldman (2003) investigated how the reliability of stimuli affects the utility of long-term memory. The authors argued that the key to deciding the evolutionary advantage of learning, is the amount of variability in the environment. They suggested that the relationship between environmental variability and the utility of learning follows “Goldilocks principle”: For learning to be beneficial, environmental variability needs to “just right” – not too high or too low. Based on their results, the authors concluded that under rapidly changing environmental conditions, a short memory span is beneficial, and that a reliable world favors using more memory. Presumably, a completely reliable world would remove the need for memory at all, as responses could be hard-wired, but the authors did not consider multiple generations of individuals, so the possibility of genetically optimized responses was not present.

Dunlap and Stephens (2009) provided the first experimental demonstration, through experiments on populations of *Drosophila melanogaster*, that some types of environ-

mental change favor learning, while others select against it. Through an aversion learning experiment, the authors were able to identify two different types of environmental change that affected the evolved degree of learning in the fruit flies differently. The first type of change, termed *best-action fixity* describes to what degree the best action to take in the environment is always the same. A high value of this parameter indicates that a strategy always performing the same action will be successful. The second type of change, termed *reliability of experience* represents the fixity of the relationship between experience and the best action. This indicates to which degree it is *possible* to do associative aversion learning. The situation that most strongly selects for learning, is the one in which there is a high reliability of experience, and a low best-action fixity. The opposite situation selects for non-learning (fixed) strategies. This theoretical model was confirmed by experiments on a population of *Drosophila* over 30 generations.

In our experiments, focus will be on regulating best-action fixity – in other words, the action giving the most fitness is subject to change, but the feedback indicating to the agent whether an action was “good” or “bad” is always correct. This ensures that the agent can always learn the best action by association with the feedback signal. We believe Dunlap and Stephens left out part of the truth: as the fixity of best-action decreases sufficiently, learning will be *selected against*. This follows from Kerr and Feldman’s hypothesis that the utility of learning follows “Goldilocks principle”.

### The Baldwin Effect - Regulating Plasticity Across Generations

The Baldwin effect (Baldwin (1896)) is an interesting example of how evolution will regulate plasticity across generations to reduce costs. The effect suggests how learned traits may become encoded into the genome of individuals through an *indirect* mechanism.

The Baldwin effect has two phases. It is initiated by a change to the environment, which forces a population to adapt. In the first phase, learning accelerates the rate of evolution. The reason is that, because learning allows weak individuals to become better, it smooths the fitness landscape, making the “evolutionary search” simpler. This was demonstrated in (Hinton and Nowlan (1987)) for an extreme case where there was only one correct solution, and no fitness gradient to steer the evolution when learning was not present. Adding learning provided a fitness gradient, accelerating evolutionary progress. In this phase, the benefits of plasticity outweigh the costs, leading to an increasingly plastic population.

In the second phase of the Baldwin effect, *genetic assimilation* occurs, meaning that the learned traits gradually become part of an individual’s genotype. This is a result of the cost of plasticity. Mayley (1996b) points out that it is the varying cost/benefit trade-off of plasticity that enforces

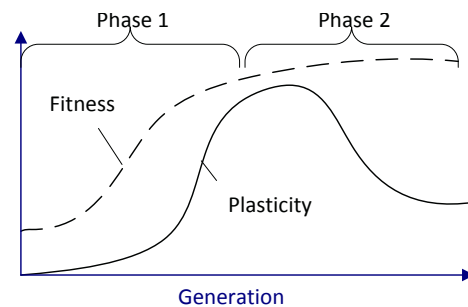


Figure 1: The two phases of the Baldwin effect - First, the benefit of learning leads to an increase in average plasticity. Subsequently, the costs lead plasticity to drop off.

the changes in the levels of learning in an evolving population: Shortly after an environmental change, the benefits of learning are large, and learning is selected for. As the population is full of individuals that can adapt to the environmental change, the cost of learning puts the individuals with innately good strategies at an advantage, and this reduces the overall plasticity in the population. Figure 1 illustrates the Baldwin effect.

**Computational modeling of the Baldwin effect** A few experiments have been done on the Baldwin Effect using simple, evolving individuals. Most of these deal with a *fixed* environment, where an unadapted population is inserted. If the environment is allowed to change during evolution, analyses become more complex. However, as argued by Anderson (1995), it is not sufficient to study the interaction between learning and evolution in fixed environments. Certain interactions between evolution and learning, e.g. the ability of plasticity to act as a “buffer” against changes in the environment, are especially evident in variable environments.

Watson et al. (2002) studied the relationship between the complexity and stability of a learning task and the tendency for genetic assimilation to occur. Genetic assimilation was found to be most complete (eliminating the most learning) when the environment was highly unstable. For more stable environments, the degree of genetic assimilation was lower. These results may seem surprising, but they follow from the relatively short periods of stability the researchers investigated. The most frequent changes were so frequent that evolved responses performed better than learned ones. A slightly more stable environment gave a higher benefit of learning, so the learning rate never reached zero.

Mayley (1996a) studied the effect of two important variables on the amount of learning performed by individuals over many generations of evolution: 1) The cost of learning, and 2) the correlation between genotype and phenotype space. He found that both a *neighborhood correlation* between genotype and phenotype space and an evolutionary *cost of learning* was necessary to observe genetic assimila-

tion.

Sasaki and Tokoro (1999) studied how rates of change in an environment affected populations of individuals evolving with different rates of *heritability of acquired characteristics*. The authors saw signs of a Baldwin Effect in the environments with relatively small degrees of dynamic change. For environments with larger degrees of dynamism, however, no genetic assimilation was seen. This indicates that the Baldwin Effect needs a certain *degree of stability* to enter into its second phase. This proposition is supported by findings presented herein.

### Sensitive Periods - Regulating Plasticity Within Individuals

A sensitive period is a period in the life of an individual where environmental stimuli have particular importance in the development of a certain ability (Knudsen (2004)). Hubel and Wiesel's classic paper (Hubel and Wiesel (1970)) illustrates the concept: One eye of a kitten was sutured in various periods throughout life, and it was found that visual deprivation of one eye early in life would make that eye unable to follow the regular path of development. The result would be that the cat was blind on that eye, also when it was opened later in life.

**Computational modeling of sensitive periods** Bullinaria (2003) studied sensitive periods of learning, as part of a simulation of the human oculomotor system. By the use of an evolutionary algorithm, *age-dependent neural plasticity* was generated. The type of age-dependent plasticity arising from these experiments had parallels with biological sensitive periods. For the purposes of our discussion, the most interesting feature of sensitive periods is their ability to reduce the cost of learning, by shrinking the plastic period of individuals. Previous studies on evolution of sensitive periods (Bullinaria (2003), Kirby and Hurford (1997)) have also discussed the cost-reducing role of sensitive periods. In this paper, we want to study this more systematically, by evolving sensitive periods under different balances between the cost and benefit of learning. Also, we want to compare the genetic assimilation happening under sensitive periods with that happening under a constant plasticity, to see if a more focused learning period can in some circumstances reduce the pressure on going through genetic assimilation.

### Hypothesis

The hypothesis of this paper is illustrated in Figure 2, and it proposes a model for how the topics of the Baldwin effect, sensitive periods and the cost/benefit balance of plasticity are connected. The way we regulate the cost/benefit of plasticity is by regulating the *rate of environmental change*. Figure 2 shows how we hypothesize learning strategies are related to the rate of environmental change. In constantly changing environments, learning has no benefits, as there

are no lasting rules to be learned. We hypothesize that the cost of learning would eliminate all plasticity in such a situation. In a fully stable environment, we also hypothesize that learning will be selected against, for obvious reasons.

In between these two extrema, we believe we will find individuals with different degrees of plasticity. Adding some slow changes to the fully stable state will at first be handled by evolution: genetic changes can tackle the environmental fluctuations. But when the environmental change reaches a certain frequency, the limit of genetic assimilation will be reached, and learning will be beneficial, as evolution cannot track the changes by itself. When the changes are relatively slow, we propose that sensitive periods may be enough to handle them, allowing short periods of plasticity in individuals' life, without paying the cost of a lifelong adaptation. Finally, in situations that have a rapid (but not too rapid) rate of environmental change, lifelong re-adaptation will be necessary, and we propose that individuals will evolve to have a high learning rate throughout life, and not just in sensitive periods. In this context, we can view sensitive periods as a *compromise* between the inexpensive but slow adaptation of the genotype and the costly but rapid adaptation allowed by individual learning.

The idea that environmental variability and evolved plasticity are closely connected has been around for a long time (Bradshaw (1965)). However, only a few empirical studies have investigated this connection (see Komers (1997) for a review), and typically only for a couple of levels of environmental variability. A systematic exploration of many scales of environmental change is naturally difficult to implement in a biological experiment. Therefore, this paper attempts to take a middle ground between theoretic approaches and empirical studies, by using evolutionary computation to evolve individuals under a large range of environmental variation.

## Experimental Setup

### The Environment

The setup is a modification of the experiment in (Todd and Miller (1991)), designed for studying the evolution of associative learning. This experiment was concerned with a simple underwater creature that is born into a patch of an environment where it spends its entire life. Substances of two different colors continuously float by, and the only decision the creature needs to make, is whether to eat these substances or not. Substances can be either poisonous or nutritious, and the challenge is for the creature to decide which type of color to consume. The association between color and edibility is a function of the feeding patch the agent was born into, so it is not optimal for all creatures to use the same strategy. The authors studied how evolution and learning together can find good strategies for this associative learning task.

In this paper, the same setup is used to study the relationship between plasticity and degree of environmental change.





Figure 2: A hypothesized spectrum mapping the level of environmental stability to learning strategies in individuals.

Two important extensions to the experiment have been performed: 1) Learning is associated with a *cost*, simulating the biological costs of learning, and 2) The environment changes at regular intervals. The second extension means that instead of varying across *feeding patches*, associations vary across *time*. The environmental change is the same: a reversal of the color/edibility association. We regulate the change rate of the environment using a single variable, the *stability period* of the environment. This variable decides the number of generations between changes. Setting this below one means we have several changes per generation. For instance a stability period of 0.1 entails 10 changes per generation.

The timing of changes within a generation is randomly chosen. However, in *which generations* the change occurs is fully controlled by the given stability period. We do not add randomness to which generations see environmental changes, because we want the generational interval between changes to be fixed, in order to study genetic assimilation systematically.

### The Agents

As pointed out by Mayley (1996b), two conditions are necessary in any experiment on genetic assimilation:

1. The plasticity of agents needs to be under genetic control.
2. The characteristics expressed by plasticity must also be possible to express genetically.

In this experiment, these conditions were met by an artificial neural network (ANN) capable of both employing *hard-coded rules* and *neuromodulated learning* when deciding which substances to eat and which to avoid. An evolutionary algorithm decides the initial *connection weights* and the *learning rates* along the same connections in the network, meaning it can evolve both hard-coded and plastic individuals.

The neural network is shown in Figure 3. The dotted connections are plastic links, which have evolvable initial weights and learning rates. The other connections are hard-wired in the experiment. The connections attaching to other connections are *neuromodulators*. That means that they modify the learning rates in the links they affect. This way, reinforcement learning driven by the perception of rewards and punishments is achieved. When the associations in the

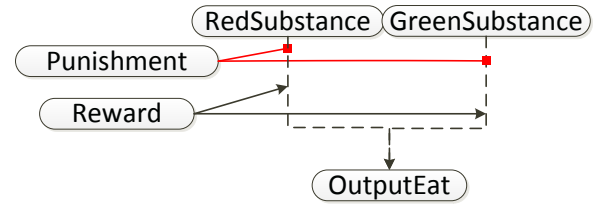


Figure 3: The neural network performing the substance association task. Rounded rectangles represent neurons. Arrows represent connections.

environment change, the agent will notice that actions lead to different reinforcing feedback than before, and alter its preferences based on the neuromodulated plasticity.

Arcs in the network are updated by the following learning rule:

$$\Delta w_{ij} = \eta * mod * |x_i x_j| \quad (1)$$

where  $\eta$  is the evolved learning rate, *mod* is the strength of incoming neuromodulation and  $x_i x_j$  is the product of pre-synaptic and post-synaptic activity, in other words a regular Hebbian update term.

As the equation shows, it is the *absolute value* of the hebbian update that is used in the calculation of the new weight value, since we want the *modulatory signal* to decide the direction of the weight change: negative modulation means whatever action was taken was a bad idea, so the weight of the link causing the action should be decreased. Positive modulation should have the opposite effect. In the absence of modulation (in other words, if  $mod = 0$ ), weights are not updated.

### Plasticity

An important question in these experiments is how the ability to employ age-dependent plasticity, potentially forming sensitive periods, affects the balance between genetic and neural adaptation. We ran the experiments with two different types of plasticity. In the first type (“static” plasticity), plasticity was constant throughout the lifetime of an individual, and regulated by a single evolved learning rate. In the second type (“dynamic” plasticity), a function was evolved, which controlled the plasticity level throughout agents’ lives with a 2 timestep interval - meaning plasticity levels could

Parameter	Value
Generations	200
Adults	15
Children	25
Crossover probability	0
Mutation probability	0.01
Genes per individual	3 (static) or 60 (dynamic)
Bits per gene	8
Elite fraction	0.1
Culling fraction	0.1

Table 1: Parameters of the Evolutionary Algorithm

change at most 50 times in the agents’ 100-step lifetimes. To produce a somewhat smooth age-plasticity mapping, the evolved function was smoothed with a window size of 8 around the current timestep. For each timestep, a value between  $-2$  and  $2$  was evolved, and this was averaged with the 7 following values to produce the plasticity value for that timestep.

### Evolutionary Algorithm

The system SEVANN was used for evolving learning rates in these experiments. SEVANN is a flexible system for designing experiments allowing evolution of neural network parameters and topologies. For details, see Downing (2010). The parameters of the evolutionary algorithm are given in Table 1.

Results were found to be most stable and evolvable when crossover was turned off. However, for adding realism to the model, investigating further how crossover affects these results is interesting for future studies. Evolved individuals employed either static or dynamic plasticity regulation, and this required a different number of evolved genes. In both cases, two of the genes coded for the innate strategy of individuals. The remaining encoded the plasticity for the rest of individuals’ lives.

## Results

### Change Rates and Learning Effort

To investigate the hypothesis illustrated in Figure 2, we evolved the learning efforts and innate weights of individuals under many different rates of environmental change. We did this both for learning rates allowed to vary throughout the lifetimes of individuals and for learning rates that remained constant throughout life.

Figure 4 shows the resulting evolved learning effort. The measured learning effort is proportional to the sum of learning efforts made in each timestep for an individual. When the learning effort is “dynamic” (in other words, allowed to vary with the age of an individual), we can identify four main learning strategies, corresponding to the four strategies we hypothesized in Figure 2. For changes occurring too

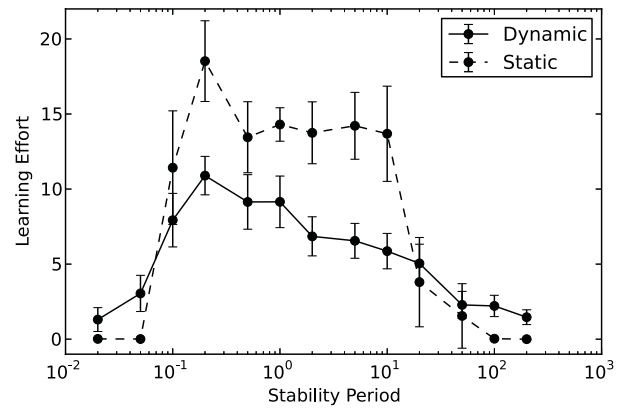


Figure 4: The learning effort made by individuals evolved under different rates of environmental change. Dots are measured values, the lines interpolate between measurements. “Dynamic” indicates a learning effort allowed to vary throughout the lifetime of individuals, while “Static” indicates a constant learning effort throughout life. Change rates are given as number of generations between each change. – Averages over 20 runs. Error bars show a 95% confidence interval of the means.

slowly or too rapidly, there is no benefit to learning. These observations are in line with the suggestion (Kerr and Feldman (2003)) that the utility of learning in a varying environment follows “Goldilocks principle”: Change rates need to be *just right* for learning to evolve.

When the change rate is within the range needed to evolve learning capacity (here, that range is from about 0.1 to 50 lifetimes between each change), we can identify two main strategies for the dynamic learners. The first main strategy is to stay plastic throughout life. This strategy is adapted for environments where change rates are so high that a sensitive period of learning would not allow an individual to keep track of environmental changes. We see this strategy evolve when there are from 1 to 10 changes per generation. Figure 5 shows the learning strategies of evolved individuals. The situation for a stability period of 1 represents a turning point, where individuals go from being plastic their whole life to having a sensitive period of learning early in life. For changes occurring less frequently than once per generation, individuals adopt the strategy of having a short period of plasticity early in life and staying non-plastic otherwise. This gives the benefit of being able to adjust to the current environment, while not incurring the cost of lifelong plasticity.

The results discussed so far are presented in a more compact form in Figure 6. This figure shows how the evolved learning efforts through life vary with the rate of environmental change. The same results were seen in Figure 4, but then without the “age”-dimension. Seeing how learn-

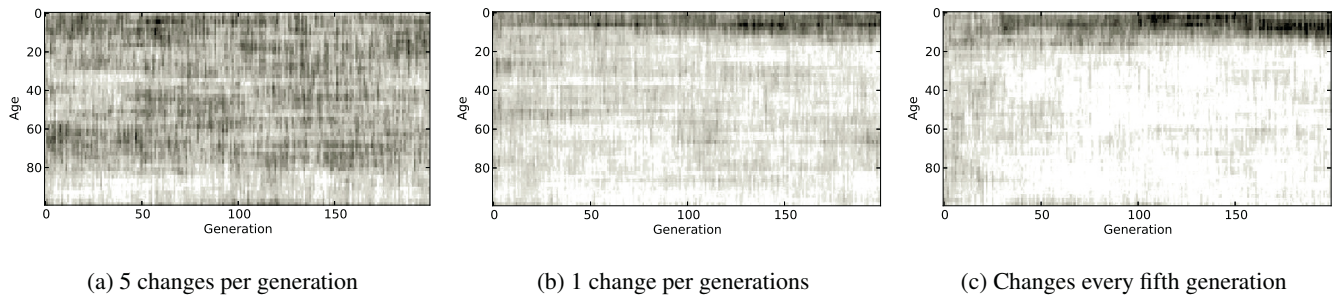


Figure 5: Average learning rate (brightness - darker means higher learning rate) through their 100-timestep lifetimes (y-axis) for the winner individuals of each generation of evolution (x-axis). – Averages over 20 runs.

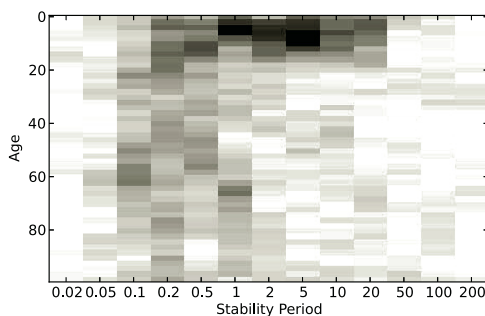


Figure 6: The learning effort made by individuals evolved under different rates of environmental change. Dark rectangles indicate a high learning rate - so a column with a few dark and many bright rectangles indicates that this individual has evolved a sensitive period of learning. Change rates are given as number of generations between each change. – Averages over 20 runs.

ing efforts vary with age and change rates, confirms the four strategies we identified earlier. For too low and too high change rates, little effort is made to learn, as seen by the bright values at the left and right edge of the plot. The relatively uniform gray columns (stability periods from 0.1 to 0.5) indicates individuals learning throughout life. Finally, the columns with stability period from 1 to 20 shows individuals with evolved sensitive periods.

### Static or Dynamic Plasticity

Comparing “dynamic” (plasticity varying throughout life) and “static” (fixed plasticity throughout life) individuals helps clarify the role of sensitive periods in plasticity when balancing the costs and benefits of learning. It reveals some of the aspects we miss by making the simplifying assumption of a lifelong, constant learning rate, as has been common in studies of the evolutionary regulation of learning efforts.

As seen in Figure 4, the most striking difference between the two types of individuals is that dynamic learners can adjust their learning rate much more smoothly to the different rates of environmental change. Static learners, on the other hand, operate in an “on/off” mode. Too rapid or too slow changes lead to individuals evolved to avoid learning. Intermediate rates of change lead to individuals that invest very much in learning. The inability of static learners to tune their learning efforts through life also means that they will have to shut off their learning ability earlier as learning becomes less beneficial. Dynamic learners, on the other hand, will retain some learning ability also under conditions that are relatively poorly suited for learning. This can be seen at the extreme ends of the spectrum of change rates, where the learning efforts of dynamic learners fall quite slowly towards zero.

Another interesting way to compare static and dynamic individuals is to look at their respective *fitness values* before and after environmental changes. This comparison indicates how well individuals balance their learning efforts to reap the benefits but avoid the costs as much as possible. Figure 7a shows fitness values for the best individuals in the *generation before* environmental change. For instance, for individuals evolved with a stability period of 50, this means the last 49 generations of their evolution happened under stable environmental conditions. We see that dynamic plasticity regulation gives a significant fitness increase when change rates are in the region from every second to every tenth generation. This is also the region where sensitive periods are most important. For higher change rates, a lifelong plasticity is most beneficial, as we saw in Figure 5. For lower change rates, no learning is most beneficial, because individuals will have *genetically assimilated* their learned traits in this generation immediately before the next change, allowing them to reduce their cost of learning.

In the generation *after* environmental change, the situation is different, as seen in Figure 7b. Individuals with a dynamically regulated learning rate now have a benefit over

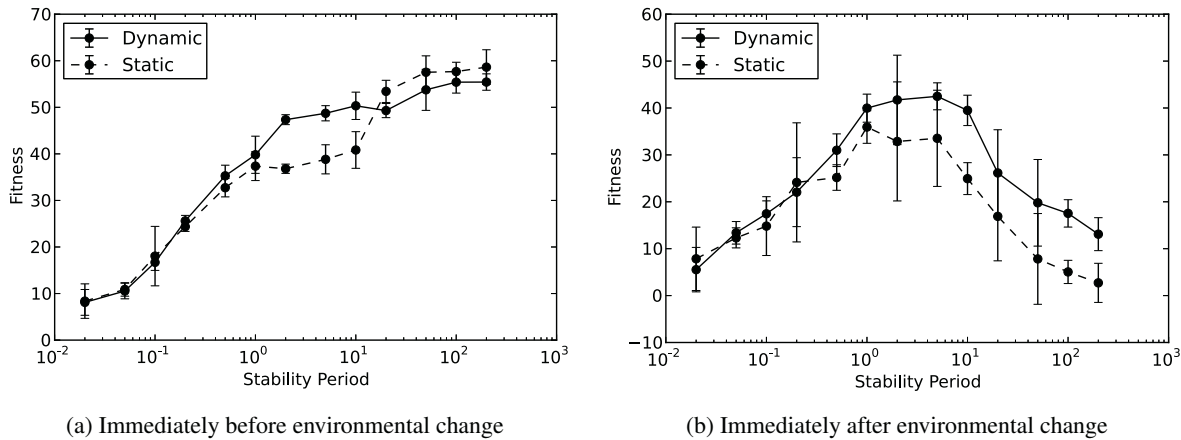


Figure 7: Graphs showing the difference in fitness values between the best evolved individuals with dynamically and statically regulated plasticities throughout life. Shown immediately before and after an environmental change. – Averages over 20 runs. Error bars show a 95% confidence interval of the means.

the static ones also when the environment has been stable for a long time. The reason is that static individuals rely more and more on genetic assimilation the longer the period of stability – meaning they are unable to respond well to the environmental change. Also notice the increase in variability between the plots, as seen in the error bars. This is a natural consequence of the environmental change: the best individual before the change will have converged to the same behavior in most runs – but their behavior in the new environment may not be the same.

The same effect can be seen by studying the fitness curves of individuals. In this case, we look at their fitness values *without* an imposed cost of plasticity. The fitness values we study here are proportional to the amount of foods the agent eats subtracted by the amount of poisons it eats – an indication of exactly how well it performs the association task. Figure 8 shows fitness values plotted over 200 generations of evolution. The sudden drops in fitness value are due to environmental changes, and the following climbs show re-adaptation of the individuals. Static individuals show much larger drops in fitness values as the environment changes, indicating that they have relied heavily on genetic assimilation of learned traits and eliminated much of their learning capacity. This is seen also in dynamic individuals, but to a much smaller degree – they are able to retain their learning ability for more generations, because it is not as costly.

For changes with a frequency above once per generation, the same pattern does not emerge. With such a high frequency of change, keeping a lifelong plasticity is beneficial, and static individuals often end up with a higher fitness as such a strategy is easier to evolve for them than for dynamic ones.

## Conclusion

By studying the evolution of learning strategies across a wide range of environmental change rates, we have observed four main strategies: For both too frequent and too infrequent changes, no learning evolves, as the cost of learning outweighs its benefits. For environmental change rates in the range suited for the evolution of learning, the two main strategies are 1) Lifelong plasticity, which is preferred when change rates are high, and 2) Sensitive periods of plasticity, which is preferred for relatively low change rates.

We have also seen that the ability to regulate plasticity through the lifetime of individuals has two important features that separate these individuals from those with a static learning rate: 1) Dynamic individuals show a less complete genetic assimilation when environmental changes are infrequent and 2) Dynamic individuals can gain the same benefit from learning while paying a lower cost when environmental changes have an intermediate frequency, by employing sensitive periods of learning.

These results illustrate that genetic assimilation and sensitive periods in learning have similar roles: reducing the cost of plasticity, while retaining its benefits. Because of their similar roles, they affect each other – for instance, sensitive periods reduce the need for genetic assimilation. Therefore, studying them together in the same model, is necessary to get a full understanding of the roles of these two phenomena.

## References

- Anderson, R. (1995). Learning and evolution: A quantitative genetics approach. *Journal of Theoretical Biology*, 175(1):89–101.
- Baldwin, M. (1896). A New Factor in Evolution. *The American Naturalist*, (30):536–553.



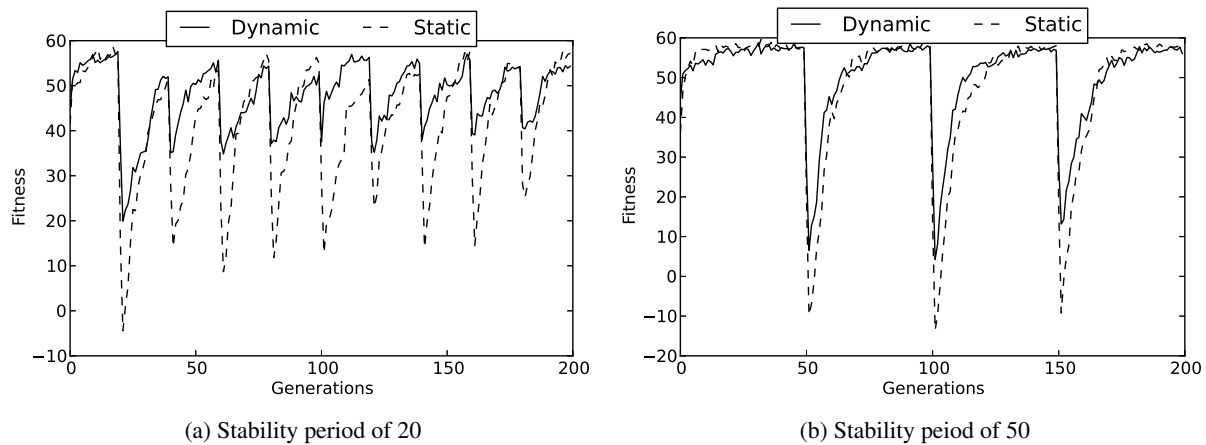


Figure 8: Fitness values plotted against generations of evolution for static and dynamic individuals. The shown fitness values are without an explicit cost to plasticity - meaning it indicates the actual performance of individuals on the association learning task. The difference between static and dynamic individuals we observe here was similar for all stability periods associated with the evolution of sensitive periods. – Averages over 20 runs.

- Bradshaw, A. (1965). Evolutionary significance of phenotypic plasticity in plants. *Advances in genetics*, 13(1):115–155.
- Bullinaria, J. A. (2003). From biological models to the evolution of robot control systems. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 361(1811):2145–64.
- Downing, K. L. (2010). A Script-Based Approach to Evolving Neural Networks. In *Proceedings of the second Norwegian Artificial Intelligence Symposium*, pages 29–36.
- Dunlap, A. S. and Stephens, D. W. (2009). Components of change in the evolution of learning and unlearned preference. *Proceedings. Biological sciences / The Royal Society*, 276(1670):3201–8.
- Floreano, D. and Urzelai, J. (2001). Evolution of Plastic Control Networks. *Autonomous Robots*, 11(3):311–317.
- Hinton, G. and Nowlan, S. (1987). How learning can guide evolution. *Complex systems*.
- Hubel, D. H. and Wiesel, T. N. (1970). The period of susceptibility to the physiological effects of unilateral eye closure in kittens. *Journal of Physiology*, 206(2):419–436.
- Kerr, B. and Feldman, M. (2003). Carving the Cognitive Niche: Optimal Learning Strategies in Homogeneous and Heterogeneous Environments. *Journal of Theoretical Biology*, 220(2):169–188.
- Kirby, S. and Hurford, J. R. (1997). The evolution of incremental learning: language, development and critical periods. Technical report, Language Evolution and Computation Research Unit, University of Edinburgh.
- Knudsen, E. I. (2004). Sensitive Periods in the Development of the Brain and Behavior. *Journal of Cognitive Neuroscience*, 16(8):1412–25.
- Komers, P. E. (1997). Behavioral Plasticity in Variable Environments. *Canadian Journal of Zoology*, 75(2):161–169.
- Littman, M. (1995). Simulations Combining Evolution and Learning. In *Adaptive Individuals in Evolving Populations: Models and Algorithms: Santa Fe Institute Studies in the Sciences of Complexity*, pages 465–477. Addison-Wesley.
- Mayley, G. (1996a). Landscapes, Learning Costs, and Genetic Assimilation. *Evolutionary Computation*, 4(3):213–234.
- Mayley, G. (1996b). The Evolutionary Cost of Learning. In *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive behavior*, pages 458–467.
- Nolfi, S., Parisi, D., and Elman, J. L. (1994). Learning and Evolution in Neural Networks. *Adaptive Behavior*, 3(1):5–28.
- Sasaki, T. and Tokoro, M. (1999). Evolving Learnable Neural Networks under Changing Environments with Various Rates of Inheritance of Acquired Characters: Comparison between Darwinian and Lamarckian Evolution. *Artificial Life*, 5(3):203–223.
- Todd, P. M. and Miller, G. F. (1991). Exploring adaptive agency II: Simulating the evolution of associative learning. In *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 306–315, Cambridge, MA. MIT Press/Bradford Books.
- Turney, P. (1996). Myths and Legends of the Baldwin Effect. In *13th International Conference on Machine Learning, Workshop on Evolutionary Computation and Machine Learning*, pages 135–142, Bari, Italy.
- Watson, J. R., Geard, N., and Wiles, J. (2002). Stability and task complexity: a neural network model of genetic assimilation. In Standish, R., Bedau, M., and Abbass, H., editors, *Proceedings of the 8th International Conference on Artificial Life (Artificial Life VIII)*, pages 153–156. MIT Press.

# Replication strategies and the evolution of cooperation by exploitation

Markus Brede<sup>1</sup> and Simon Tudge<sup>1</sup>

<sup>1</sup>University of Southampton, Southampton, SO17 1BJ  
markus.brede@soton.ac.uk

## Abstract

Introducing the concept of replication strategies this paper studies the evolution of cooperation in populations of agents whose offspring follow a social strategy that is determined by a parent's replication strategy. Importantly, social and replication strategies may differ, thus allowing parents to construct their own social niche, defined by the behaviour of their offspring. We analyse the co-evolution of social and replication strategies in well-mixed and spatial populations. In well-mixed populations, cooperation-supporting equilibria can only exist if the transmission processes of social strategies and replication strategies are completely separate. In space, cooperation can evolve without complete separation of the timescales at which both strategy traits are propagated. Cooperation then evolves through the presence of offspring-exploiting defectors whose presence and spatial arrangement can shield clusters of pure cooperators.

## Introduction

Actions that are in the interest of the group but not necessarily to the immediate benefit of the individual are widely observed in the social and biological sciences. Understanding the emergence and sustainability of such altruism or cooperation still poses major challenges to evolutionary game theory and the recent decades have seen very active research in this field. For instance, a recent review article classified five different mechanisms that support altruism (Nowak, 2006). Here, we are mainly interested in one of them: network reciprocity, cf. (Szabó and Fath, 2007; Perc and Szolnoki, 2010) for recent reviews.

Network reciprocity summarizes effects that result from constrained interactions in structured populations in which agents interact with fixed and typically rather small sets of permanent neighbours. In this way interactions between parents and offspring are favoured, i.e. cooperation is supported through positive assortment of strategies. The literature about evolutionary games in structured populations goes back to the seminal paper of Nowak (Nowak and M., 1992) in which spatial games were introduced, observing and describing chaotic patterns of strategies in space. The work was extended in several ways to, e.g., include effects

of noise (Szabó and Toke, 1998) and asynchronicity (Huberman and Glance, 1993) in strategy propagation. Recent research has mainly focused on the evolution of cooperation in population structures modelled by complex networks, finding, e.g. that heterogeneous networks give a strong boost to cooperation (Santos et al., 2006). The latter findings have been extended to evolutionary models on regular graphs in which there is some heterogeneity in agent's abilities to generate payoff. Examples of studies in this direction are (Szolnoki and Szabó, 2007; Perc and Szolnoki, 2008; Brede, 2011a), but also the recent work on teaching and learning (Szolnoki and Perc, 2008; Tanimoto and Yamauchi, 2012). In the latter line of research agents are classified into two groups: (i) teacher agents with an enhanced ability to pass on strategies and (ii) learner agents with reduced abilities to pass on strategies. The co-evolutionary dynamics of fast and slow strategy spread can then generate phases in which cooperation can survive much beyond parameter regimes in which cooperation is supported by network reciprocity alone (Brede, 2013a).

Common to this large bulk of work on cooperation and network reciprocity is the assumption that offspring (in a biological context) or followers (in a social context) adopt exactly the same strategy as parents (or leaders). In fact, one might surmise that this assumption is crucial to allow for positive assortment which enables support for cooperation through network reciprocity. In this paper we introduce a more general framework that aims to challenge this hypothesis and explore its boundaries. We distinguish two traits that describe agent behaviour. The first is the typical social strategy that describes an agent's behaviour in the social dilemma game under consideration. The second is a replication strategy, i.e. a strategy that an agent will pass on as a social strategy to its offspring. In this way every agent is characterised by a tuple  $(s, e)$ : a social strategy  $s$  and a replication strategy  $e$  through which it can determine its offsprings' social behaviour. Notably, the social strategy and the replication strategy of an agent can be different: It might be in the interest of an agent to surround itself with offspring (or followers) that are of a different type than itself. Hence,

agents may surround themselves by un-like types, questioning the role of positive assortment by network reciprocity.

One might also interpret our framework as a very simple model of social niche construction (Powers et al., 2011). The term social niche construction was recently introduced to describe a situation in which agents can evolve preferences for the social group they interact with. Using the example of preferences for group size it was then demonstrated that co-evolution of such preferences and social strategies can naturally support cooperation. In our context here, by their replication strategy, agents can influence the environment in which they live and thus improve their chances to generate payoff in competitive games. Using the often-studied framework of the prisoner’s dilemma game, we will explore under which circumstances such a simple co-evolutionary model can allow for additional support for cooperative strategies.

Real-world inspiration for the above assumption of differences between social and replication strategies is not hard to come by. For instance, in models of teaching and learning the above framework allows for situations in which teachers can teach strategies different from their own. Arguably, this is a more realistic and general framework than the one considered in previous work. In a biological context one might interpret the model as a simple model of cell differentiation.

The present work thus follows in a line of recent advances in the understanding of the co-evolution of individual-level traits and cooperation (Szolnoki et al., 2009; Powers et al., 2011; Perc and Wang, 2010; Brede, 2011b, 2013b).

The organization of the paper will be as follows. We start with a detailed description of the model framework and then describe and explain results in the section thereafter. The paper concludes by a summary and discussion section that puts our main results into context and discusses implications and future work.

## Model

In more detail, we consider the following model of an evolutionary one-off prisoner’s dilemma in space. A set of  $N$  agents are associated with the sites of a graph whose links define interactions and directions of strategy propagation. In case of experiments in well-mixed populations this social networks is a complete graph, otherwise we perform experiments on an  $L \times L$  square lattice with von Neumann neighbourhoods. Agents are characterized by two strategy traits, a social strategy trait  $s \in \{0, 1\}$  and a replication strategy trait  $e \in \{0, 1\}$ . We use the convention that state “0” corresponds to a strategy of pure defect and state “1” corresponds to pure cooperate. Agents play a prisoner’s dilemma with payoff matrix parametrized in the conventional form

$$\begin{pmatrix} R & S \\ T & P \end{pmatrix} = \begin{pmatrix} 1 & -r \\ 1+r & 0 \end{pmatrix}, \quad (1)$$

such that the parameter  $r$  characterizes the toughness of the game setting. In Eq. (1)  $R$  stands for the reward for mutual

cooperation,  $S$  for the “sucker’s” payoff,  $T$  for the temptation to defect, and  $P$  for the punishment for mutual defection. A small  $r \ll 1$  corresponds to very mild dilemma settings, whereas  $r \rightarrow 1$  characterizes very tough dilemmas. Hence, we distinguish four strategies: (i) cooperators who want their offspring to cooperate ( $s = 1, e = 1$ ), (ii) cooperators who want their offspring to defect ( $s = 1, e = 0$ ), (iii) defectors who want their offspring to cooperate ( $s = 0, e = 1$ ) and (iv) defectors who pass on defection to their neighbours ( $s = 0, e = 0$ ). This model may easily be extended by including context-dependent inheritance, i.e. the offspring determining trait would then depend on the social strategy currently played, but we reserve a thorough investigation of this case for future work and concentrate on the simplest setup in this paper.

In the following we will also consider the impact of various timescales in the evolution of social and replication strategies. The spread of both strategies might occur on separate or similar timescales. In the case of joint strategy pass, an agent will adopt the desired social strategy of a parent as well as its replication strategy. In case of disparate pass, an agent might either adopt the parents’ desired social strategy or its replication strategy. To distinguish these cases and to investigate the effects of disjoint strategy pass we introduce a probabilistic framework for the spread of strategies: With probability  $p_s$  only the social strategy is imposed, otherwise, with probability  $p_a$  only the replication strategy is passed on, and in the remaining cases (i.e. with probability  $p_p = 1 - p_s - (1 - p_s)p_a$ ) the social strategy is imposed and the replication strategy passed on. The timescales of the spread of social and replication strategies are then given by  $T_s = 1/(p_s + p_p)$  and  $T_a = 1/(p_a + p_p)$ .

Hence, our evolutionary simulations consist of an asynchronous process iterating the following steps:

- Seed all agents with randomly chosen initial social and replication strategies.
- Randomly pick a focus agent, say  $i$ , and choose a reference agent  $j$  from one of its four von Neumann neighbours at random.
- Evaluate game interactions of the focus agent with its neighbours to determine its accumulated payoff  $\pi_i^{(\text{game})}$  and follow the same procedure to calculate the accumulated payoff  $\pi_j^{(\text{game})}$  of the reference agent from interactions with its neighbours.
- After evaluating game payoffs, a cost  $c$  is deducted from payoffs of agents who attempt to spread a strategy different from their social strategy, i.e.

$$\pi_i = \pi_i^{(\text{game})} - c(1 - \delta_{s_i e_i}), \quad (2)$$

where  $\delta_{ij} = 1$  if  $i = j$  and 0 otherwise. A cost  $c > 0$  accounts for the fact that imposing social strategies different

from your own might involve a costly effort to ‘convince’ the opponent. Unless otherwise stated experiments are carried out with  $c = 0$  and the influence of a non-zero cost is only evaluated at the end of the paper.

- In a next step, a focus  $i$  agent will adopt the strategy of the reference agent  $j$  with a likelihood that depends on the difference in payoffs, i.e.

$$P(j \rightarrow i) = \frac{\exp(\pi_j/\kappa)}{\exp(\pi_j/\kappa) + \exp(\pi_i/\kappa)}. \quad (3)$$

In the above equation the parameter  $\kappa$  introduces noise in the replication process, the larger  $\kappa$ , the larger the chance for inferior strategies to spread. In all following simulations we set the noise level to a relatively large value of  $\kappa = 1$ . This choice is motivated by reasons of computational feasibility, because the evolutionary dynamics becomes very slow for low levels of noise when the timescales of cluster expansion are dominated by the timescales of change of local configurations of  $s = 0, e = 1$  defectors surrounded by cooperators at the boundaries of clusters of pure cooperators/defectors which can become entrenched for a very long time (see also the results section).

- Strategy spread (with the probability  $P(j \rightarrow i)$  defined above) occurs in the following way. With probability  $p_p$  the reference agent will impose his desired social strategy and will also transfer his replication strategy ( $s_i = e_j$  and  $e_i = e_j$ ). Otherwise, with probability  $p_s$  only the social strategy is imposed ( $s_i = e_j$ ) and in the remainder of cases, i.e. with probability  $p_a = \frac{1-p_p-p_s}{1-p_s}$  only the replication strategy is passed on from  $j$  to  $i$  ( $e_i = e_j$ ). The timescales for joint or disjoint spread of the traits (parametrized via  $p_p$ ) and distinction of timescales for the spread of social and replication strategies (parametrized via  $p_s$ ) prove crucial parameters to understand the dynamics of social evolution in this context.
- The payoff evaluation and strategy updating steps are then repeated for a sufficiently large number of steps until a quasistationary state has been reached. Then, average concentrations of all strategy concentrations are sampled from another  $TN$  iterations (note that in this paper time is always measured in units of full lattice sweeps).

In the following we employ computer simulations of systems composed of in between  $10^4$  and  $1.6 \times 10^5$  agents to construct phase diagrams of parameter regions in which the evolutionary dynamics can allow cooperation to survive.

## Results

### Well mixed populations

Before discussing spatial simulations it is worthwhile analysing the case without network reciprocity, i.e. a well-mixed population in which individuals meet at random and

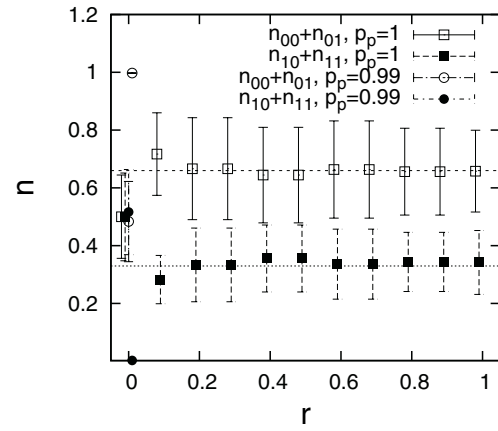


Figure 1: Dependence of the concentrations of defect and cooperate strategies on dilemma toughness for a well-mixed population of size  $N = 40000$  and noise level  $\kappa = 0.01$ . For  $p_p = 1$  cooperation can always survive, but for  $p_p < 1$  defection wins out for  $r > 0$  (and since they all overlap circles at  $n = 0$  are omitted for  $r > 0.01$ ).

strategies spread according to Eq. (3) on the basis of payoff gathered from interactions with the entire population. For simplicity, we will not distinguish timescales given by  $p_s$  and  $p_a$  and assume  $p_s/p_a = 1$  in the following. It is then straightforward to describe the evolutionary dynamics of strategy concentrations  $n_i$  by a set of rate equations in the form:

$$\dot{n}_i = \sum_{k,l} n_k n_l a_{kl}^{(i)}, \quad (4)$$

where the indices label the four possible strategies 00, 10, 01, and 11 and the matrices  $a^{(i)}$  contain information about conversions between strategies according to the rules set out in the previous section. It is worth noting that  $n_{00} + n_{10} + n_{01} + n_{11} = 1$ , i.e. there are only three relevant degrees of freedom.

For the transition matrices one finds:

$$a^{(00)} = \begin{pmatrix} 0 & 0 & -1/2 & -p \\ \beta P & \beta/2 & 0 & 0 \\ \beta/2 & 0 & \beta p & 0 \\ \alpha P & \alpha/2 & 0 & 0 \end{pmatrix}, \quad (5)$$

where we introduce the shortcuts  $\beta = 1 - p_p/2$ ,  $\alpha = 1 - p_p$ ,  $P = 1/(1 + \exp(-\Delta\pi/\kappa))$ , and  $p = 1/(1 + \exp(\Delta\pi/\kappa))$  to simplify notation with payoff difference between defectors and cooperators

$$\Delta\pi = n_C(1 + r) - (-(1 - n_C)r + (1 - n_C)) \quad (6)$$

$$= r, \quad (7)$$

where  $n_C = n_{11} + n_{10}$  is the concentrations of agents with social strategy cooperate.



For instance, if an agent with strategy 00 meets an agent with strategy 01, the agent following 00 will adapt its strategy with likelihood  $1/2$  (since both strategies achieved the same payoff). If strategy 00 learns from 01, there are three cases that need to be distinguished. (i) with probability  $1 - p_p$  00 learns the social strategy that 01 wishes to impose (i.e. 1) and 01's replication strategy (i.e. 1) and hence converts to strategy 11. (ii) with probability  $p_p/2$  00 only learns the social strategy 01 wishes to impose, i.e. 00 converts to 10 and (iii) 00 may only learn 01's replication strategy, i.e. 00 converts to 01. In all three cases 00 converts to a strategy different from 00, hence the entry  $a_{13}^{(00)} = -1/2$ . Similarly, if 10 encounters 00 the chance that 10 will learn from 00 is given by  $P$ . Either learning only the social strategy 00 wishes to impose (probability  $p_p/2$ ) or learning both the social strategy 00 wishes to impose and 00's replication strategy (probability  $1 - p_p$ ) will convert 10 to 00. Hence the entry  $a_{21}^{(00)} = \beta/2$ .

Analogous equations for the remaining three matrices  $a^{(10)}$ ,  $a^{(01)}$ , and  $a^{(11)}$  can be derived, i.e.

$$a^{(10)} = \begin{pmatrix} 0 & 0 & p_p/4 & pp_p/2 \\ -\beta P & -\beta/2 & -\beta P & -\beta/2 \\ 0 & 0 & 0 & 0 \\ Pp_p/2 & p_p/4 & 0 & 0 \end{pmatrix}, \quad (8)$$

$$a^{(01)} = \begin{pmatrix} 0 & 0 & p_p/4 & pp_p/2 \\ 0 & 0 & 0 & 0 \\ -\beta/2 & -\beta p & -\beta/2 & -\beta p \\ Pp_p/2 & p_p/4 & 0 & 0 \end{pmatrix}, \quad (9)$$

$$a^{(11)} = \begin{pmatrix} 0 & 0 & \alpha/2 & p\alpha \\ 0 & 0 & p\beta & \beta/2 \\ 0 & 0 & \beta/2 & \beta p \\ -P & -1/2 & 0 & 0 \end{pmatrix}. \quad (10)$$

The systems of equations (4) is a system of three non-linear equations. Even though an analytical analysis of stationary states might be possible, numerical integration of (4) provides enough insight for the present purposes. Figure 1 gives the dependence of stationary strategy concentrations obtained by numerical integration of (4) on the dilemma strength for two scenarios of strategy pass for  $\kappa = 0.01$  (note that noise levels should be measured per interaction, i.e. a very small value in the well-mixed case with all-to-all interactions corresponds to larger noise values on sparse grids).

The first, illustrated by square symbols in Fig. 1 corresponds to completely asynchronous strategy pass, i.e.  $p_p = 1$ . In this case for all  $r > 0$  the population is split into roughly two thirds defectors (equal halves of which carry both replication strategies) and one third cooperators (with again equal halves carrying both replication strategies). In

contrast, for any  $p_p \neq 1$  (round symbols) the social strategy cooperate is found to die out, i.e.  $n_{10} + n_{11} = 0$ , and the two social defect strategies share the population in equal proportions.

It is easy to understand why this is the case. Strategy  $s = 0$  and  $e = 1$  can earn the same payoff as pure defectors with  $s = 0$  and  $e = 0$ . However, in a well-mixed population it cannot profit from generating offspring who cooperate, because cooperation can be exploited by the entire population of defectors. Hence, agents with  $s = 0$ ,  $e = 1$  can generate the same number of 'offspring' as  $s = 0$ ,  $e = 0$ ; but their descendants die out without conferring an advantage on the parents. The situation is different if the spread of social strategy and replication strategy are completely separated: In this case the population of social cooperators is always reinforced by an inflow from the pool of social defectors with replication strategy defect (who earn equal payoff as pure defectors) and can also not be suppressed through interactions with pure defectors, because there is always a one-half chance that the cooperative trait survives due to separate strategy pass.

### Spatially distributed populations

As we have seen in the previous section on well-mixed populations, replication strategies that differ from social strategies can only support cooperation if the spread of social and replication strategies is completely separated. The reason for this is that social niche construction cannot operate in well-mixed populations: offspring that plays the social strategy cooperate can be exploited by the entire population and does not bestow any specific benefit to the parent who gave birth to it. Rather, the effect for strategies with replication strategy  $e = 1$  is negative: Their offspring will replicate less well than the parent because it can be exploited by the entire population of defectors. One would anticipate that this situation can be different in viscous populations. In the latter case, parents can accrue specific individual fitness benefits by surrounding themselves by cooperators. It appears reasonable to surmise that the consequential increase in reproductive fitness for parents might compensate for the loss in fitness of offspring, thus enabling cooperative strategy traits to survive. We will explore this scenario for spatial games in some detail below.

Figure 2 illustrates simulation results for the evolution of the four strategies in two typical settings in which replication of the two components of a strategy, social strategy  $s$  and replication strategy  $e$ , are to some extent disjoint ( $p_p = 0.6$ ). The figures also give the frequency  $f_c$  of mutually cooperative interactions. In the first setting with lower dilemma toughness (top panel), cooperation can grow to dominance. In the second with somewhat larger dilemma toughness (bottom panel), an equilibrium state in which all four strategies coexist is reached. Spatial arrangements of the strategies that correspond to such a mixed state are illustrated in Fig.

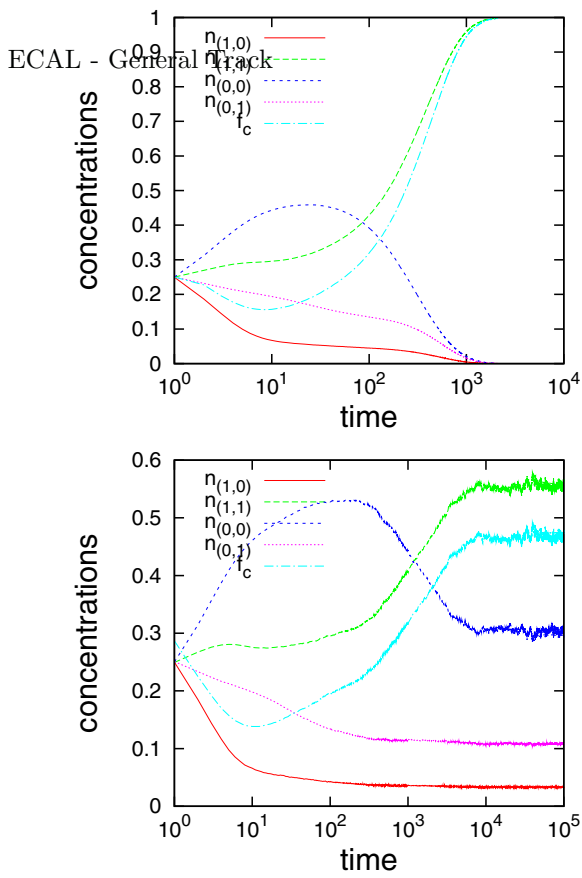


Figure 2: (Average) evolution of social and reproduction strategies for a prisoner's dilemma and average fraction of mutually cooperative interactions  $f_c$ . (a) With  $r = 0.18$  and  $p_p = 0.6$  when cooperation grows to dominance and (b) with  $r = 0.22$  when an equilibrium between the strategies is reached (on a  $200 \times 200$  torus with  $\kappa = 1$ ).

3.

These first experiments which we show in Fig. 2 illustrate two important points: (i) As hypothesised above, when including opportunities for social niche construction via replication strategies, cooperation can survive in spatial arrangements, even if strategy pass is not completely disjoint. (ii) Disjoint transmission of social and replication strategies can allow for the dominance of cooperation in regimes of dilemma games far beyond regimes normally supported by network reciprocity (i.e. for a typical spatial game with von Neumann neighbourhoods with  $\kappa = 0.1$  the extinction threshold for cooperation is around  $r_c = 0.021$  (Hauert and Szabo, 2004) and even somewhat smaller with  $r_c \approx 0.017$  for  $\kappa = 1$ ).

The typical spatial arrangements in Fig. 3 also provide an intuitive understanding why replication strategies can support cooperation in spatial settings. The figure shows the

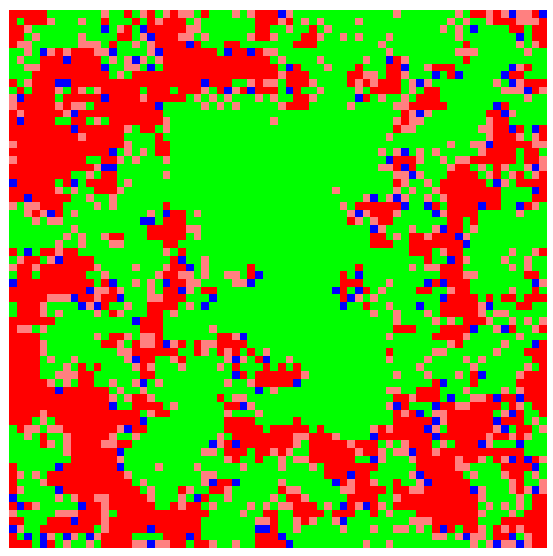


Figure 3: Example configuration of an equilibrium arrangement of the four strategies (for  $r = 0.22, p_p = 0.6, \kappa = 1$ ). Colors are  $(s, e)$ : red (D,D), light red (D,C), green (C,C) and blue (C,D). (70\*70) C,D and D,C only occur at the boundaries of larger C,C and D,D clusters.

presence of large homogeneous clusters of pure defectors ( $s = 0, e = 0$ , dark red) and pure cooperators ( $s = 1, e = 1$ , green). Strategies with  $s \neq e$  only occur at the boundaries of these clusters. A cursory glance at Fig. 3 which is confirmed by the results shown in the bottom panel of Fig. 2 also suggests that the strategy  $s = 0, e = 1$  (blue) is far more prominent than strategy  $s = 1, e = 0$ . The reason is that a social defect strategy can earn larger payoffs than a social cooperate strategy.

Let us now consider the effect of  $s = 1, e = 0$  and  $s = 0, e = 1$  on the clusters of pure cooperators and defectors. When replicating in the direction of pure cooperators  $s = 1, e = 0$  either reproduces itself or (assuming disjoint strategy pass) it produces a defector with  $s = 0, e = 1$ . However, since  $s = 1, e = 0$  earns the same payoff as pure cooperation  $s = 1, e = 1$  can only invade clusters of cooperators via neutral drift. By the same token it only rarely gets a chance to replicate when competing against defection, and if so, it cannot reproduce itself (since any pure defector  $n$  would either only be influenced in its social strategy via the replication strategy, i.e.  $s_n = e = 0$  or would additionally imitate the replication strategy  $e_n = e = 0$  which correspond to its own strategy anyway). Hence,  $s = 1, e = 0$  impedes the spread of pure cooperation into clusters of pure defectors, but also, by transitioning into  $s = 0, e = 1$ , delays invasions of defection into clusters of pure cooperators.

What about the spread of the strategy  $s = 0, e = 1$ ? The propagation of  $s = 0, e = 1$  is more relevant at the boundary of clusters, since, following social defect, it will typi-

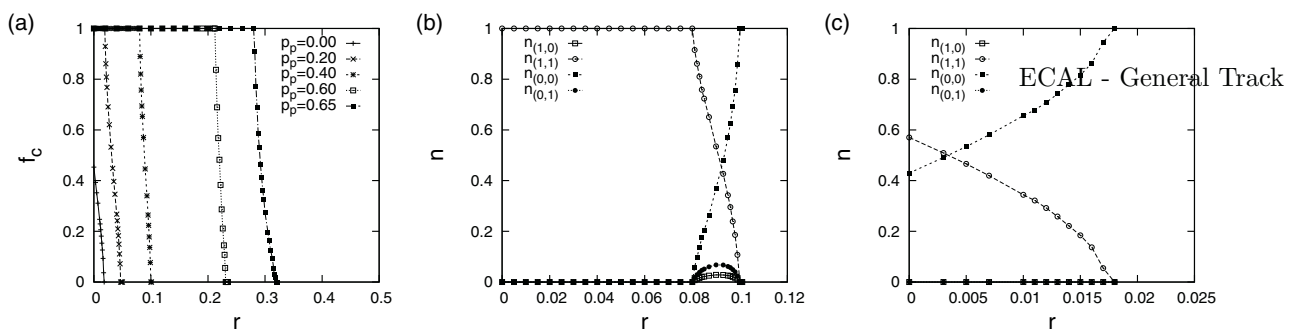


Figure 4: (a) Dependence of the frequency of mutually cooperative interactions  $f_c$  on the dilemma strength for various values of  $p_p$  on a  $200 \times 200$  torus. It becomes apparent that cooperation finds more and more support, the more frequent uncorrelated replication events become. (b) Dependence of the concentrations of the various strategies on the dilemma strength for  $p_p = 0.4$ . There are two phases dominated by pure cooperators or pure defectors (small and large  $r$  and an in-between phase in which all strategies can coexist). (c) In contrast, for  $p_p = 0$  only pure cooperators and pure defectors survive and the phase diagram from the standard PD without replication strategies is reproduced.

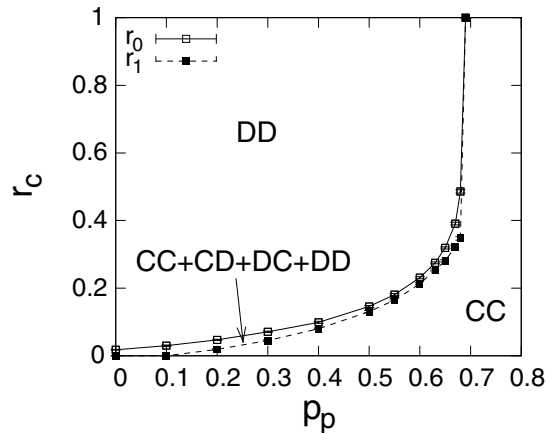


Figure 5: Dependence of critical thresholds for the extinction of strategies on the probability for disjoint trait propagation  $p_p$ .

cally harvest a larger payoff than  $s = 1, e = 0$ . On the one hand, when interacting with pure cooperation, it will always surround itself by pure cooperation. On the other hand, when interacting with a pure defector, it will either generate a  $s = 1, e = 0$  defector or cause a transition of the neighbour to pure cooperation. Hence, even though  $s = 0, e = 1$  exploits cooperators, it also shields clusters of cooperators from the invasion of defection and promotes the spread of the pure cooperate strategy.

When considering the role of all four strategies at the boundaries of compact clusters of pure cooperators and pure defectors it is also important to recognize that the strategy  $s = 0, e = 1$  will typically generate the largest payoff (because being on average surrounded by more cooperators than pure defect) and thus replicate most often. Even though

being thus most successful in terms of replication, it can only recreate itself indirectly – its offspring will never follow the same strategy.

The mechanism which supports cooperation in the simulations shown above principally works as follows. Offspring-exploiting defectors  $s = 0, e = 1$  are the most successful strategy, but cannot recreate directly, and, as a result, serve as support for cooperation. It is evident that in case of joint strategy propagation a 'checkerboard pattern' of  $s = 0, e = 1$  interspersed with  $s = 1, e = 1$  would be evolutionarily stable. However, since  $s = 0, e = 1$  cannot recreate itself and is only generated at boundaries of defectors and cooperators when disjoint strategy pass is allowed, without the presence of random strategy invasions or mutations such a pattern cannot evolve from random initial conditions (cf. Fig. 4 right panel). Moreover, this checkerboard pattern is not stable in the face of even small degrees of disjoint strategy spread (measured by  $p_p$ ). If  $p_p$  is sufficiently large, offspring-exploiting defectors support pure cooperation in two ways: (i) by shielding clusters of pure cooperators from the invasion of defection, and by (ii) serving as a source of pure cooperators, as a consequence of their own success in replication.

Figure 5 extends our earlier simulation experiments by giving the dependence of the frequency of cooperative interactions and strategy concentrations on the dilemma toughness. A clear dependence of the support for cooperation on the frequency of joint strategy pass  $p_p$  is evident and is further supported by the full phase diagram that illustrates the dependence of coexistence regimes and regimes in which single strategies dominate on  $p_p$ . As already indicated by the dependencies in Fig. 4 the coexistence regimes are typically rather small and regimes in which either pure cooperation or pure defection take over the entire population dominate the diagram of Fig. 5. Coexistence is only found in a small

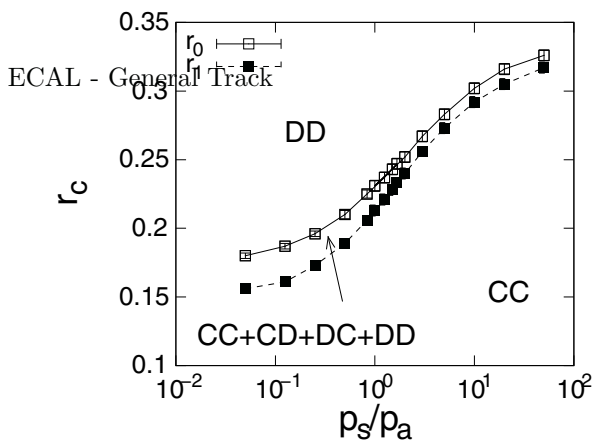


Figure 6: Dependence of critical thresholds for the extinction of strategies on timescales for the spread of the social strategy ( $p_s$ ) and for the replication strategy ( $p_a$ ) for fixed  $p_p = 0.6$ . For  $r > r_0$  only pure defection survives, for  $r < r_1$  only pure cooperation survives and for  $r_0 > r > r_1$  all four strategies can coexist. The faster social strategies spread relative to replication strategies, the more support for cooperation. Also the coexistence region becomes larger the slower the spread of the social strategy.

borderline region between the regimes of pure strategy dominance.

It is also of interest to investigate the dependence of the support for cooperation on the relative timescales for strategy propagation. To explore this question, we set up experiments with a fixed frequency of disjoint strategy pass and vary the relative frequencies with which reference agents only impose their replication strategy as the desired social strategy of neighbours (i.e. with probability  $p_s$ ) and the frequency with which neighbours only learn the replication strategy of a reference (i.e. with probability  $p_a$ ). A typical phase diagram that summarizes our simulation experiments is given in Fig. 6.

Two observations stand out. First, the support for cooperation grows the more prominent the imposition of social strategies relative to the learning of replication strategies. This observation is consistent with our argument about the role of offspring-exploiting defectors. The more frequent the spread of social strategies, the more often they will generate pure cooperators. Second, the regime in which all four strategies can coexist becomes the larger the more frequently agents pick up the replication strategies of their references. This second finding is also intuitively clear from the same argument. The more often replication strategies are learnt, the more often  $s = 0, e = 1$  transitions into  $s = 1, e = 0$ , thus boosting the concentrations of other strategies.

A last point worth investigating is the role of a cost for strategy imposition. To investigate this point we presume

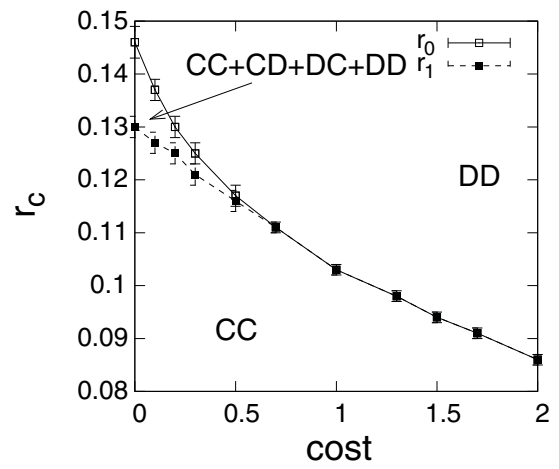


Figure 7: Dependence of critical thresholds for the extinction of strategies on costs for the propagation of unequal strategy traits  $s \neq e$  for  $p_p = 0.5$ .

that the behaviour in the standard game, i.e. imposing agents' own social strategy on neighbours, is free. In contrast, imposing a strategy different from an agent's social strategy needs "convincing", i.e. it comes at some cost  $c$ , cf. Eq. (2). The experiments carried out in this way allow us to test the stability of the standard framework and answer the question "Would differences between social and replication strategies evolve if teaching is costly?". To explore this question we fix the frequency  $p_p$  of disjoint trait transmission and assume that both traits spread at equal rates (i.e.  $p_s = p_a$ ). Figure 7 summarizes these experiments by presenting a phase diagram for the dependence of extinction thresholds on cost assumptions. Clearly, imposing a cost for producing offspring with social strategies different from an agent's social strategy reduces support for cooperation. Such behaviour would naturally be expected, since imposing costs penalizes the "mixed" strategies  $s = 0, e = 1$  and  $s = 1, e = 0$  and our previous argument relied on the presence of the first of those to support cooperation via the exploitation of offspring. Nevertheless, even imposing costs that are very substantial compared to game payoffs, cooperation can exist in regimes far beyond the support it would find from the network reciprocity of the spatial grid (with an extinction threshold of  $r_c \approx 0.017$ , see Fig. 4).

From the data presented in Fig. 7 it is also noteworthy that costs reduce support for the coexistence regime. Costs penalize strategies with  $s \neq e$  and hence mixed phases in which all four strategies can co-exist are increasingly suppressed the larger the costs imposed.

## Conclusions

In this paper we have introduced a simple way to explore social niche construction (Powers et al., 2011) on spatial networks. In our framework, on top of a social strategy every



agent is endowed with a second trait, a replication strategy, which allows the agent to determine the social strategy of its offspring. We then explored the co-evolution of social and replication strategies, subject to various assumptions about the timescales of spread of both strategy components.

Analyzing the dynamics of the co-evolution in the prisoner's dilemma, we have established that cooperation can only be supported in well-mixed populations if social and replication strategies are never both passed on from parent to offspring. In a social context this corresponds to the rather unrealistic assumption that the timescales of learning the respective traits are completely separated. In a biological context, this assumption translates into assumptions about the traits being located on uncoupled separate genes. As demonstrated by our exploration of the spatial prisoner's dilemma, the presence of a structured population can mitigate this strict condition. We have shown that in spatial settings cooperation can find very strong support, even if the simultaneous passing on of social and replication strategies is rather likely. The main driver of the support for cooperation is the prevalence of offspring-exploiting defectors which can generate the largest payoffs in the game. Offspring-exploiting defectors are found to be in a similar role as payoff-distinguished agents in (Perc and Szolnoki, 2008; Brede, 2011a): by virtue of their enhanced ability to pass on strategies they assume a "leadership" role (Zimmermann and Eguíluz, 2005). Different from previous models like (Zimmermann and Eguíluz, 2005; Perc and Szolnoki, 2008; Brede, 2011a), however, such agents with  $s \neq e$  never replicate identically and thus offspring-exploiting cooperators facilitate the spread of cooperation by surrounding themselves with cooperators.

We have also presented a number of further experiments that corroborate the robustness of the above finding. Support for cooperation is robust to changes of the timescales of strategy spread over several orders of magnitude and also the inclusion of substantial costs for imposing social strategies different from an agent's own social strategy do not alter outcomes in a qualitative way.

One may wonder if the framework in which we introduced replication strategies in this paper is too restrictive. In other words: Would our main findings be robust if replication strategies were context dependent, i.e. influenced by the social strategy of agents which replicate such that an agent in the role of a social cooperator may wish to impose a different strategy on its neighbours than when being in the role of a social defector? We reserve a more comprehensive analysis of this more general setting for future work.

## Acknowledgements

The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

## References

- Brede, M. (2011a). The evolution of cooperation on correlated payoff landscapes. *Artificial Life*, 17:365–373.
- Brede, M. (2011b). Playing against the fittest: A simple strategy that promotes the emergence of cooperation. *EPL*, 94:30003.
- Brede, M. (2013a). Costly advertising and the evolution of cooperation. *PLoS ONE*, available online: <http://dx.plos.org/10.1371/journal.pone.0067056>.
- Brede, M. (2013b). Short versus long term benefits and the evolution of cooperation in the prisoner's dilemma game. *PLoS ONE*, 8(2):e56016.
- Hauert, C. and Szabo, G. (2004). Game theory and physics. *Am. J. Phys.*, 73:405–414.
- Huberman, B. A. and Glance, N. S. (1993). Evolutionary games and computer simulations. *Proc. Natl. Acad. Sci. USA*, 90:7716–7718.
- Nowak, M. A. (2006). Five rules for the evolution of cooperation. *Science*, 314:1560–1563.
- Nowak, M. A. and M., M. R. (1992). Evolutionary games and spatial chaos. *Nature*, 359:826–829.
- Perc, M. and Szolnoki, A. (2008). Social diversity and promotion of cooperation in the spatial prisoner's dilemma game. *Phys. Rev. E*, 77:011904.
- Perc, M. and Szolnoki, A. (2010). Coevolutionary games - a mini review. *BioSystems*, 99:109–125.
- Perc, M. and Wang, Z. (2010). Heterogenous aspirations promote cooperation in the prisoner's dilemma game. *PLoS ONE*, 5:e15117.
- Powers, S. T., Penn, A. S., and Watson, R. A. (2011). The concurrent evolution of cooperation and the population structures that support it. *Evolution*, 65:1527–1543.
- Santos, F. C., Rodrigues, J., and Pacheco, J. (2006). Graph topology plays a determinant role in the evolution of cooperation. *Proc. R. Soc. B*, 273:51–55.
- Szabó, G. and Fath, G. (2007). Evolutionary games on graphs. *Phys. Rep.*, 446:97–216.
- Szabó, G. and Toke, C. (1998). Evolutionary prisoner's dilemma game on a square lattice. *Phys. Rev. E*, 58:69–73.
- Szolnoki, A. and Perc, M. (2008). Coevolution of teaching activity promotes cooperation. *New J. Phys.*, 10:043036.
- Szolnoki, A. and Szabó, G. (2007). Cooperation enhanced by inhomogeneous activity of teaching for evolutionary prisoner's dilemma game. *EPL*, 77:30004.
- Szolnoki, A., Vukov, J., and Szabo, G. (2009). Selection of noise level in strategy adoption for spatial social dilemmas. *Phys. Rev. E*, 80:056112.
- Tanimoto, J. and Brede, M. and Yamauchi, A. (2012). Network reciprocity by coexisting learning and teaching strategies. *Phys. Rev. E*, 85:032101.
- Zimmermann, M. and Eguíluz, V. M. (2005). Cooperation, social networks, and the emergence of leadership in a prisoner's dilemma game with adaptive local interactions. *Phys. Rev. E*, 72:056118.

# When congestion can be useful: modelling driver diversion behaviour in road traffic networks

James R. Snowdon<sup>1</sup>, Ben Waterson<sup>2</sup>

<sup>1,2</sup> University of Southampton, UK

<sup>1</sup>jrs105@soton.ac.uk

## Abstract

The ability to accurately predict driver route choices is an important part of traffic assignment, the process of forecasting traffic flows on roads across a region. Many assignment methods only consider the presence of recurrent forms of congestion, such as during rush hour periods, and fail to incorporate non-recurrent congestion effects caused by irregular events such as road traffic accidents. This paper proposes an agent based driver route choice model which includes driver reactions to the presence of non-recurrent congestion, supposing that drivers learn relationships between congestion locations and adjust their expectation of network travel times en-route, potentially choosing to divert. By simulating an example network with mixed populations consisting of agents capable of diverting and not, the result is found that initially increasing the proportion of diverting agents from zero is beneficial to the system as might be expected, reducing the number of vehicles navigating the incident affected area, but beyond a tipping point agents can no longer perceive the presence of congestion prior to diverting and network performance decreases. The model not only demonstrates the conflict between agents adopting travel time reducing behaviour and its impact on system performance, but it also highlights the importance of modelling driver knowledge appropriately to reproduce plausible phenomena in simulation.

## Introduction

It is important to be able to predict the impacts on road traffic flows of potential interventions such as new road layouts or signal timings, construction of new homes and retail zones, or population growth over time. To assist with this prediction, which can suggest severity of congestion or levels of pollution, traffic assignment is the methodology which attempts to answer the question ‘what is the likelihood that drivers will use this route to travel from an origin point to a destination point?’. After establishing an expected level of vehicular demand between each origin and destination pair, the congestion causing feedback effect of many driver’s routing decisions is included in the assignment process.

In order to predict road traffic flows adequately a thorough understanding of how drivers make routing decisions is required. This can then be translated in to models of driver route choice behaviour which describe how relative

measures of route attractiveness inform final routing decisions, typically including key preference factors such as travel time, distance travelled or toll costs (Bekhor et al., 2006). Capturing the important motivations and other elements of driver choice, such as reactions toward the presence of congestion, is crucial in order to produce accurate models and subsequent traffic flow predictions.

Two types of congestion are generally considered to exist in road traffic networks. The first is congestion caused by vehicular demand exceeding capacity (in the form of maximum traffic flow) on a regular basis, such as during rush hour and other peak periods, known as ‘recurrent’ congestion, and is present (in some form) in most assignment methods. The second ‘non-recurrent’ form of congestion is caused by unexpected capacity reducing incidents such as accidents, road maintenance works or local surges in demand reducing excess capacity, and rarely features in practical traffic assignment applications. This work examines the extent to which the possible presence of non-recurrent congestion could influence traffic flows and accordingly the importance of considering it in traffic assignment.

Traditional assignment approaches adopt an aggregate approach to analysis, splitting overall demand according to the relative attractiveness or utility of whole route options with ‘usual’ road network properties and so ignoring any possibility of non-recurrent congestion. The recent rise of agent based models of route choice instead allow modelled drivers to make individual decisions based on their unique experience of the transportation network, allowing for learning strategies and the evolution of choice over time to be explored (Nagel and Marchal, 2006).

This paper examines a mechanism whereby agents, each representing single drivers, are able to react to new information regarding the presence of non-recurrent congestion en-route. If this ability provides agents with a different impression of road network characteristics causing them to act in a different but plausible manner to what would usually be predicted, then it can be considered an important aspect of driver choice which should be included in mainstream traffic assignment models.

The paper is structured as follows: firstly the current state of, and assumptions behind, traffic assignment methods and driver route choice models are provided. Then the underlying traffic interaction model used here is described followed by the novel behaviour model presented in this work. The behaviour model is implemented on an simple example road network and the outcomes and insight provided by this model are finally discussed.

### Modelling traffic flows at user equilibrium

Traffic flows between each origin and destination pair under consideration are generally assumed in predictions to be in a configuration known as ‘user equilibrium’, originally defined by Wardrop (1952). Here the generalised travel costs (disutility) to travellers of each used route is minimal and equal to any other, thus providing zero incentive for a driver to use a different route. Extensions to the definition of user equilibrium have been made since including ‘Dynamic User Equilibrium’, where travel times are equal and minimal in each departure time period (Peeta and Ziliaskopoulos, 2001), and ‘Stochastic User Equilibrium’, where a portion of travellers choose sub-optimal routes representing driver perception errors (Bekhor et al., 2006).

An aspect of route flow dynamics which has received relatively little attention from modelling research is the process by which transportation networks move to an equilibrium flow set from any initial flow configuration, such as when a network change occurs and route flows pass through disequilibrium states. Some models have been developed, such as a class of ‘day to day’ route choice models derived from the work of Horowitz (1984) which assume that daily traffic flow configuration is a function of flows on the previous day(s) which can tend towards an asymptotic equilibrium flow distribution as the number of simulated days tends to infinity. These models have been extended to consider the basins of attraction of multiple flow equilibria (Bie and Lo, 2010) and long term system behaviour (Smith et al., 2013). Crucially for this work, day to day route choice models have also been developed incorporating multi agent systems where each agent, representing a single driver, holds a unique, evolving impression of network attributes which guide future route choice decisions (Liu and Huang, 2007; Tiang et al., 2010). These methods have been found to give identical predicted traffic flow configurations to other assignment approaches (Snowdon, 2013).

### Driver reaction to non-recurrent congestion

Where non-recurrent congestion is considered in traffic assignment, it is generally represented as a capacity reduction occurring along a section of road which increases the travel time for a constant level of vehicular demand (Gao et al., 2008). To model drivers diverting to avoid congestion, Unnikrishnan and Waller (2009) introduce the concept of ‘re-course’ where drivers are modelled receiving up to date net-

work state information as they traverse the network, which is a technique also adopted by other traffic simulation tools (Sykes, 2010). In reality however many drivers will not be privy to such information and must rely on previous experience to guide network travel time expectations.

Modelling driver re-routing and its influence on route choice has previously been attempted in network representations which impose a spatial congestion structure externally to the model (Gao et al., 2008). In reality the location of queues move within-day as a result of driver routing decisions (Long et al., 2008), for example the consequence of drivers diverting to avoid non-recurrent congestion may introduce more non-recurrent congestion elsewhere in the network and clear the area initially afflicted. It is important for accurately modelling traffic flows that the outcome of drivers adopting diversion behaviour, and any impacts of potential diverting opportunities on initial route choices, is understood.

This work incorporates two main features in the traffic assignment process in order to capture both the effect of incomplete network information and of emergent spatial congestion structure as has been described. The first is the use of a cell transmission model, which models vehicle movements along road links and the build up of queues in parts of the network, and the second is a novel application of a coupled hidden markov model representation of driver knowledge given to agents traversing the simulated network.

### A cell transmission model of road traffic interactions and congestion propagation

In this work vehicle movements and interactions are captured using a Cell Transmission Model (CTM). The CTM used here is a reimplement of previous works (Long et al., 2008, 2011), extended from Daganzo’s original CTM (Daganzo, 1994). In this formulation a network  $G = (N, A, C)$  features a set of nodes  $N$  connected by a set of links  $A$  and includes a set of centroids  $C$  which are each attached to a single node  $n \in N$  and can represent origins and/ or destinations. In the CTM each link is discretized in to homogeneous cells and time is partitioned in to intervals such that the cell length is equal to the distance travelled by free-flow traffic in one time interval  $\delta$ . For example, here  $\delta = 5s$  and free flow vehicle speed ( $v$ ) is 15m/s so cell length is 75m. A time variable,  $t$ , advances by 5s at each simulation time step. Each cell also has a fixed capacity of vehicles which can reside inside it at each time step. The units traversing the cell transmission model are implemented as agents which each hold their own driver behaviour model.

At each time step, as well as advancing agents on to their next cells and links, a number of agents drawn from the population may be entered in to the simulation at centroids specified by the agents. Should an agent be unable to join the link connected to  $n$  they will join a centroid waiting queue and attempt to enter the network at each time step onwards. Each

simulated day agents are injected in to the simulation in the same order but the number of agents entering at each time step is randomly chosen (up to a limit of 5 per origin).

Unlike in previous implementations of the CTM, in this system each modelled agent holds a unique identity and attached behaviour model. Prior to arriving at a node, agents are interrogated for their turning intention and next link choice. The simulated day only comes to an end when all vehicles have left the network.

Equilibrium is hard to define in a stochastic traffic system since a degree of route choice ‘noise’ may ensure that flows never converge to a single stable set. When describing deterministic route choice systems, Bie and Lo (2010) define equilibrium as a route flow configuration which re-generates itself indefinitely. Here a heuristic method is adopted that the system is left to evolve for a period longer than is required to visually reach an equilibrium flow distribution.

### Modelling a driver agent’s network knowledge using a coupled hidden markov model

Previous work has reported positively on describing a period of observed road link conditions as belonging to a set of states including ‘free flow’, ‘mildly congested’ and ‘highly congested’ travel time distributions (Kwon and Murphy, 2000; He et al., 2006). In reality link states exhibit an often predictable spatial structure around the road network as queues propagate from a single starting point such as a busy junction or incident location and affect other regions. These correlations have been used as a basis for developing reliable travel time predicting algorithms (Min and Wynter, 2011).

Choosing the right number of states to represent road link performance and capture travel time variation sufficiently is not a trivial task. The appropriate set of states may vary by location and time of day under consideration. In their work, Kwon and Murphy (2000) use two states, free flow and congested, but this work uses three states to emphasise the distinction between minor and severe congestion. Drivers may experience any state regardless of its cause, but non-recurrent congestion may result in an unusual state set being experienced compared to under ‘usual’ recurrent congestion. States as used here only depend upon travel times alone:

**Free flow** Travel time on link is less than or equal to  $1.3 \cdot (\text{Free flow travel time on link})$

**Moderate congestion** Travel time on link is less than or equal to  $2.0 \cdot (\text{Free flow travel time on link})$  and greater than  $1.3 \cdot (\text{Free flow travel time on link})$

**Heavy congestion** Travel time on link is greater than  $2.0 \cdot (\text{Free flow travel time on link})$

The driver behaviour model proposed here focusses on allowing driver agents to learn link state structures through

the use of a Coupled Hidden Markov Model (CHMM). By learning link state structures drivers can re-evaluate the expectation of congestion elsewhere in the network based on that day’s experience. Traffic systems have previously been represented as CHMMs for predictive purposes (Kwon and Murphy, 2000; Herring et al., 2010) but have yet to be explored as the basis of an agent based driver knowledge representation.

### Model overview

A single hidden markov model (HMM) considers time as discrete and at each step can be in one of a number of unobservable (hidden) states,  $S$ . At each time step the model emits one of a number of externally observable symbols,  $V$ , with a given probability in each internal state,  $B = \{b_j(k)\}$   $j \in S, k \in V$ . A transition probability distribution describes the state which the model will be in at the next step given the current state,  $C = \{c_{ij}\}$   $i, j \in S$ . The probability of the model being in any initial state is given by a distribution,  $\pi = \{\pi_i\}, i \in S$ .

This work uses an extension to HMMs proposed by Zhong and Ghosh (2001) which allows for the consideration a network of *coupled* hidden markov models. The state of each HMM in the next time step is influenced not only by itself but also by the (hidden) states of other connected HMMs. Here the CHMM is achieved by extending both the system transition matrix to describe the influencing effect of model  $a'$  on model  $a$ ,  $C = \{c_{ij}^{(a',a)}\}$ , initial probability distributions,  $\pi = \{\pi_i^a\}$ , and also introducing a coupling matrix  $\Theta = \{\theta_{a',a}\}$  which defines how the set of HMMs are connected.

### Implementation details

Each driver agent is equipped with a single CHMM as described, with each HMM representing a single road link in the network. The goal of the model is to determine an expectation of travel time for a link  $a$ ,  $\varphi^a$ , both at the start of each simulated day and en-route once the state of other links elsewhere in the network has been observed. For this implementation the CHMM belonging to each driver agent is considered fully connected ( $\Theta = J_{|L|}$ ) although in their work Kwon and Murphy (2000) only connect HMMs considered connected in the road network. The relationships between nearby link states are not fully understood yet limiting the interconnectedness of the CHMM saves on computations and memory use required by the simulation.

Driver agents also store an associated expected travel time (in simulation steps, i.e. multiples of 5 seconds) for each state of each link,  $\Gamma = \{\gamma_j^a\}, j \in S, a \in A$ , which is adjusted by daily experience using the exponentially weighted moving average model where  $r^a$  is the experienced travel time on link  $a$  and  $\alpha$  is an externally set learning parameter (0.01 in this implementation):



$$\gamma^a = \alpha r^a + (1 - \alpha)\gamma^a \quad (1)$$

Once a driver agent has traversed a link  $a$  they observe the link's state exactly (i.e. each state is tied to only one observation,  $B = I_{|S|}$ ) so only the state in question's expected travel time is updated. The initial state distribution,  $\pi^a$ , is then updated directly as the average experienced proportion of occasions that the link was in state  $s$ . At the start of the simulation the probability of any link being in any state is equal and each state's expected travel time,  $\gamma_j^a$  = free flow travel time on link.

The expectation of travel time on network links at the start of any simulated day can then be simply found as  $\varphi = \Gamma\pi$ .

This model would be sufficient to ignore within day effects and determine an equilibrium set of route flows based on initial route choices alone. However this work sets out to incorporate the possibility of agents changing the expectation of a link's state en-route based on information relating to the state of other links obtained on the trip, within day.

The system transition matrix is updated at the end of each simulated trip,  $C = \{c_{ij}^{(a',a)}\}$ , as the experienced proportion of occasions that link relationships occurred. That is, for the HMM associated with link  $a$ , the state probability distribution describes the probability of link  $a'$  being in state  $j$  given that link  $a$  was in state  $i$ .

As an agent travels through the network, experience is accumulated in two sets which are re-initialised as empty at the start of each day;  $L = \{a\}$ , which stores the identifiers of experienced links, and  $O = \{o^a\}$  which stores the corresponding set of observed link states. This information is used to update the expected probability that link  $a$  will be in state  $i$ ,  $P(o_i^a)$  as the average expected state of link  $a$  given its relationships with each of the traversed links in the set  $L$ :

$$P(o_i^a) = \frac{\sum_{(a' \in L)} c_{a',a}^{o_i^a}}{|L|} \quad (2)$$

Thus the single expected travel time on link  $a$  can be re-evaluated en-route as:

$$\varphi^a = \sum_{i \in S} P(o_i^a) \cdot \varphi_i^a \quad (3)$$

To inform the choice model, the final utility of a route is given as  $\varphi \cdot -0.1$  (negative since travel time is a disutility). The discrete choice model used to determine the probability of an agent choosing a route is a path sized logit model (with calibrated parameter = 0.1) which takes in to account the overlap between route options as well as utility (Bekhor et al., 2006).

In this application of a CHMM the re-evaluation of HMM states occurs when the agent receives any new information about current link states, such as when leaving a link. The result of this is that the CHMM does not operate in fixed

time steps which would create unnecessary computations or information not being considered in network re-evaluations.

## A simulation of driver reaction to network incidents

As an illustration of the effects of incorporating the general CHMM agent knowledge representation, the proposed day to day traffic assignment method is performed on the network shown in figure 1 featuring a fixed vehicular demand of 7500 vehicles on each simulated day. The network consists of 13 links, 13 nodes and one origin to destination pair O0-D0. Each link is divided in to 10 cells with a capacity of 10 vehicles with two exceptions: link 6, which is divided in to 16 cells, and link 5, which is divided in to 3 cells.



Figure 1: The network structure under examination, also showing the cells associated with links and vehicles traversing the network in a non incident affected day.

The probability of all network cells being affected by incidents on each day is 0% except cell 9 on link 6, whose capacity drops from 10 to 3 when an incident occurs in the same manner as is shown in figure 2 which illustrates how congestion forms in the CTM. On any simulated day there is a 30% chance that link 9 cell 6 is perturbed in this way. All other constants are set as in the model definition.

There are two routes through the network here forward named as the 'major route' and 'diversion route' which consist of links  $[1, 2, 3, 4, 5, 6, 13]$  and  $[1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13]$  respectively. The free flow travel times on the two routes are therefore 345 seconds

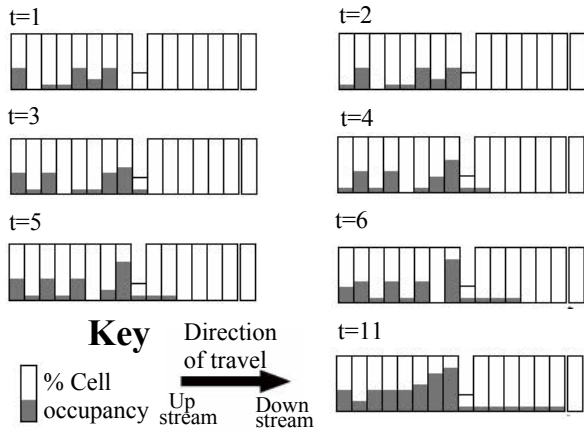


Figure 2: Cell transmission model output showing a stream of vehicles encountering a cell of lower capacity, resulting in the formation of an upstream queue.

on the major route (5.2km) and 565 seconds on the diversion route (8.5km).

Although the CHMM knowledge model presented here suggests that all agents should re-evaluate their decision en-route, in reality not all drivers will be able to do so either due to personal reluctance or lack of knowledge regarding the area and alternative route options. Accordingly, each agent holds the CHMM behaviour model as described except for the following key differences: *switchers* will re-evaluate their route choices en-route and may choose to divert (although it is important to remember that not all will, the discrete choice model only provides a likelihood of choosing a route rather than a decision) and *stayers* who will not re-evaluate any system perceptions en-route. As described in the literature overview, the vast majority of transportation forecasting models do not consider that agents will process information en-route, thus consist of a population of 100% stayers (0% switchers) who might be armed with perfect congestion information pre-trip.

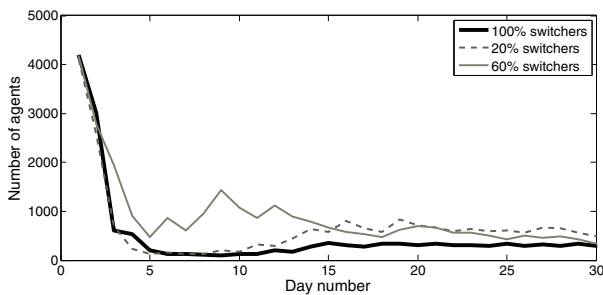


Figure 3: Numbers of agents choosing the diversion route as their initial route choice against simulation day.

Figure 3 shows the day to day initial route choices of 7500 agents traversing the network in figure 1 over a period of 30 days. The population consisting of 100% switchers is capable of moving to an equilibrium which features fewer agents initially choosing the diversion route. This can be considered modelling the ability for switchers to ‘take a chance’ on the preferable major route and stayers being forced to consider average network performance in their routing decisions so more often initially choosing the diversion route.

As an analysis of within-day system behaviour, figure 4 shows the proportion of the agent population engaged in switching against time during a single incident affected day once route flows reached equilibrium (beyond day 50). The diverting proportion is calculated as the number of agents which have chosen to switch routes and are present on links 7, 8, 9, 10, 11 and 12 (the diversion section) against the number of agents present on links 6, 7, 8, 9, 10, 11 and 12 (the combination of the diverting section and incident affected link). As figure 4 shows, the 7500 agents take between 15000 seconds ( $\approx 4$  hours) and 30000 seconds ( $\approx 8$  hours) to pass through the network for the strategy mixtures examined.

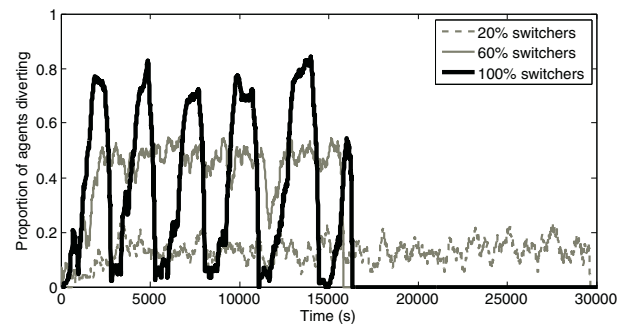


Figure 4: Proportion of agents engaged in diverting throughout an incident affected simulated day at equilibrium.

As would be expected, for lower percentages of switchers present in the population the proportion of agents engaged in switching is capped by that percentage. It is logical that if few switchers exist in the population each will engage in diverting, enjoying a reduced travel time of close to free flow conditions on the diversion route. Due to the discrete choice model used, if an agent predicts that link 6 has a higher travel time than at the start of their journey, it only becomes *more likely* that it will divert, hence not every switcher agent chooses to divert.

The trend of ‘maximum numbers of switching agents divert’ would not be expected to continue with increasing the proportion of switchers in the population. If a population of 100% switchers exists and the maximum number divert then the major incident affected route would hold a close to free flow travel time and thus be faster than the diversion route.

Figure 5 summarises and extends figure 4, showing the

proportion of only the switching agents engaged in diverting during a 12500 second portion ( $\approx 3.5$  hours) of the simulated day for varying proportions of switchers. Below 60% switcher populations, the mean proportion of switchers engaged in diverting reaches 0.77, with the standard deviation decreasing to 0.09. Beyond 60% a clear system change appears in both figures 4 and 5 as the average proportion of agents diverting falls. There are two reasons for this; first the mechanism as described above suggests that some switching agents will choose not to divert - although this is few as figure 4 shows that the maximum proportion of agents diverting in a population of 100% switchers is close to 0.77. Secondly, the periodic ‘wave’ like diversion behaviour visible in figure 4 appears and at 100% switchers the proportion of agents switching is rarely steady as diversion behaviour regularly breaks down.

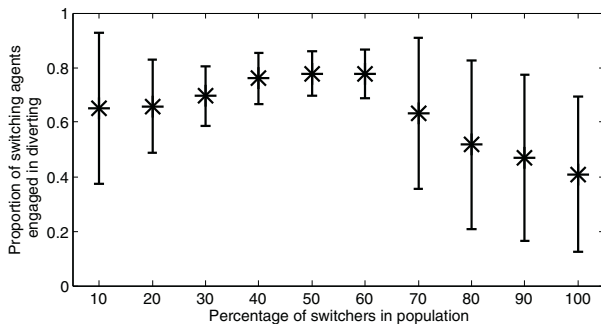


Figure 5: Mean and standard deviation of proportion of switching agents engaged in diverting between  $t = 2500$ s and  $t = 15000$ s.

To understand the impact of these trends on agent experience, figure 6 charts the average travel times between the route divergence point at the end of link 5 and route merge point at the beginning of link 13 (the same region examined by figure 4) experienced by switcher, stayer and *all* agents on an incident affected equilibrium day. This shows the (average) benefit to agents of adopting the two strategies. As has been discussed, prior to 60% switchers within the population, switcher agents enjoy a lower average travel time as stayer agents traverse the incident affected major route. Beyond 60%, the value of this benefit to switchers decreases even though in every simulation it is on average better for agents to adopt the switcher behaviour.

To examine the effect of varying population proportions on system performance, figure 7 charts the time required for all 7500 agents to pass through the network. As the proportion of switching agents in the population increases, up to around 70-80% switchers, the amount of time required falls, suggesting that the system can be considered to be acting in a *more* optimal fashion. Beyond 80% switchers in the population, despite the simulation consisting of more agents capable of making en-route diverting decisions in the hope of

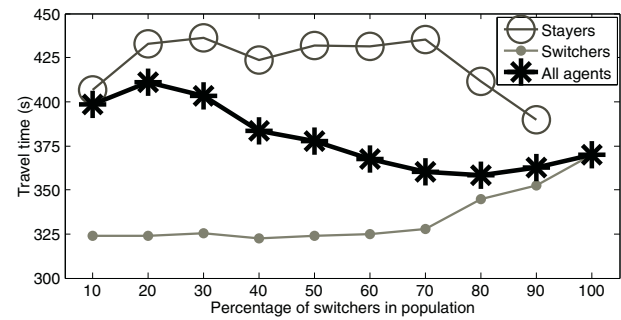


Figure 6: Mean of travel times between diverging and merging points experienced by switcher, stayer and all agents during a single incident affected equilibrium day for varying proportions of switchers in the population.

decreasing overall travel time, the time taken for all agents to traverse the network *increases*.

The graph in figure 7 also shows the ‘optimal’ time required to complete an incident affected day from a simulation where the likelihood of link 6 being affected by incidents is certain. This simulation length is lower because uncertainty is removed from the system and network attributes do not change in the day to day model. Agents can each optimise their initial route choices through the evolutionary route adaptation process and at equilibrium do not need to alter route choices within day. Consequently the negative diversion breakdown does not occur.

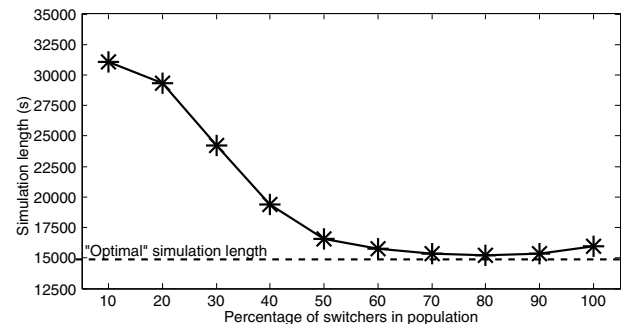


Figure 7: Time required to complete the movement of 7500 agents at equilibrium on an incident affected day. Also shown is the ‘optimal’ time required when the probability of an incident on link 6 cell 9 is 1.0.

### Diversion breakdown and the role of information

The simulation result in figure 4 has shown how, for higher proportions of switching agents existing within the population, when an incident arises the number of agents engaging in diverting rises and falls in a wave like motion which has a negative impact on overall network performance.

To demonstrate how this trend arises, figure 4 shows a series of simulation outputs at six time steps on an incident

afflicted day (as shown by the capacity decrease on link 6). In a) ( $t = 1755s$ ) agents are joining the network and, due to the existence of congestion on the links leading up to the route diverging point, perceive link 6 to be in a highly congested state. At this early point in the simulation the upstream queue is still forming and few agents are diverting. In b) ( $t = 3990s$ ) the congestion stretches up the network but due to a larger number of agents engaging in diverting the size of the queue in each cell decreases - as in the peaks in figure 4. By c) ( $t = 5040s$ ) the reduced congestion on preceding links means that agents are no longer capable of considering link 6 to be in a highly congested state even though link 6 remains affected by the incident. At d) ( $t = 5530s$ ) few agents are engaged in diverting and most agents join link 6 believing it to be clear as in the troughs from figure 4. In e) ( $t = 5930s$ ) queues re-form on links preceding link 6 and by f) ( $t = 6490s$ ) agents once again perceive that link 6 is in the heavily congested state and again engage in diverting as in the peaks in figure 4.

The simulation has shown how, when an incident occurs, agents anticipate its presence and more agents which are capable of diverting do so, the queue on preceding links decreases and agents joining the simulation receive no information about any queues occurring ahead, so are unable to predict that link 6 is in a highly congested state. Thus all incoming agents naively remain on the major route believing it clear, eventually creating more queues which then back up the carriageway restarting the cycle.

Diversion breakdown has the result of decreasing network performance despite being caused by agents trying to decrease their own travel times, suggesting that in this simulation some level of queueing on upstream links can be seen as positive. In order to reduce average travel times in a network without information provided, some drivers are required to wait in congestion so that others can benefit from observing the presence of queues.

## Conclusions

This model has demonstrated a plausible road traffic phenomena in the form of diversion breakdown which is created in simulation by incorporating within the model both inter-vehicle interactions and a driver knowledge representation which focuses on experience gathered within trip and relationships between anticipated link states.

Although many other models of driver route choice exist, as well as models which explore other aspects of driver behaviour, this work has sought to explore the consequences of a single type of behaviour - adopting en-route diversions. Scope exists to incorporate the findings of this work with existing and future driver route choice models.

The findings from this model are also more relevant to real world road networks where users can be considered to be experienced. For example in areas where drivers are not expected to possess local knowledge, such as on major road,

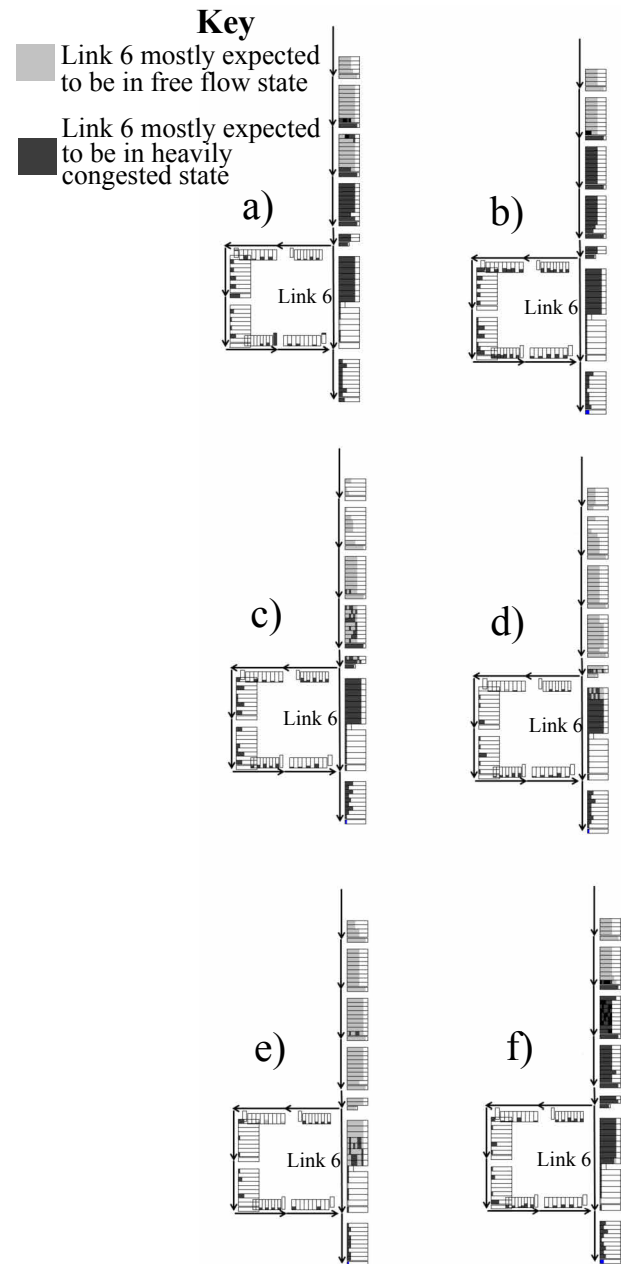


Figure 8: Simulation outputs at a number of time steps exploring system behaviour on an incident affected day at equilibrium with a 100% switchers population. Agents occupying each cell are coloured according to their belief of the current state of link 6.

they may be reluctant to divert. Additionally, a driver unfamiliar with an area may hold their own potentially incorrect assumptions regarding traffic flows which will influence their routing and en-route diversion behaviour.

Network structure will also play a key role in whether and when diversion breakdown occurs and there may be multi-



ple opportunities for drivers to divert. Additionally, overall network performance is only improved if the diversion route can accommodate increased volumes of traffic which is uncertain, even unlikely, in most real world traffic networks.

To summarise, this paper has presented a plausible and general agent behaviour model of driver road network perceptions. Through the modelling of road conditions as belonging to one of a set number of states, a coupled hidden markov model can model the relationships between states and provide expectations of driver behaviour. The simulation has shown the competing pressures on drivers as choosing to remain on their initial route choices or be open to diverting on to alternatives.

## References

- Bekhor, S., Ben-Akiva, M., and Ramming, S. (2006). Evaluation of choice set generation algorithms for route choice models. *Annals of Operations Research*, 144(1):235–247.
- Bie, J. and Lo, H. K. (2010). Stability and attraction domains of traffic equilibria in a day-to-day dynamic system formulation. *Transportation Research Part B*, 44(2):163–168.
- Daganzo, C. F. (1994). The cell transmission model: a simple dynamic representation of highway traffic. *Transportation Research Part B*, 28(4):269–287.
- Gao, S., Frejinger, E., and Ben-Akiva, M. (2008). Adaptive route choice models in stochastic time-dependent networks. *Transportation Research Record 2085*, pages 136–143.
- He, R., Kornhauser, A. L., and Yelinek, M. (2006). A study on temporal and spatial variability of travel time. In *Joint International Conference on Computing and Decision Making in Civil and Building Engineering*.
- Herring, R., Hofleitner, A., Abbeel, P., and Bayen, A. (2010). Estimating arterial traffic conditions using sparse probe data. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 929–936. IEEE.
- Horowitz, J. L. (1984). The stability of stochastic equilibrium in a two-link transportation network. *Transportation Research Part B*, 18(1):13–28.
- Kwon, J. and Murphy, K. (2000). Modeling freeway traffic with coupled hmms. *Univ. California, Berkeley, CA, Tech. Rep.*
- Liu, T. L. and Huang, H. J. (2007). Multi-agent simulation on day-to-day route choice behaviour. In *Third International Conference on Natural Computation*, volume 5, pages 492–498.
- Long, J., Gao, Z., Ren, H., and Lian, A. (2008). Urban traffic congestion propagation and bottleneck identification. *Science in China Series F: Information Sciences*, 51:948–964.
- Long, J., Gao, Z., Zhao, X., Lian, A., and Orenstein, P. (2011). Urban traffic jam simulation based on the cell transmission model. *Networks and Spatial Economics*, 11(1):43–64.
- Min, W. and Wynter, L. (2011). Real-time road traffic prediction with spatio-temporal correlations. *Transportation Research C: Emerging Technologies*, 19(4):606–616.
- Nagel, K. and Marchal, F. (2006). *Moving through nets: The physical and social dimensions of travel*, chapter Computational methods for multi-agent simulations of travel behavior, pages 131–188. Elsevier, Oxford.
- Peeta, S. and Ziliaskopoulos, A. K. (2001). Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics*, 1(3-4):233–265.
- Smith, M., Hazelton, M. L., Lo, H. K., Cantarella, G. E., and Watling, D. P. (2013). The long term behaviour of day-to-day traffic assignment models. *Transportmetrica A: Transport Science*.
- Snowdon, J. (2013). Convergence and equilibrium analysis under strategic route choice behaviour. In *UTSG: 45th Annual Conference of the Universities' Transport Study Group Meeting*.
- Sykes, P. (2010). *Fundamentals of Traffic Simulation*, chapter Traffic Simulation with Paramics. Springer.
- Tiang, L., Huang, H. J., and Liu, T. L. (2010). Day-to-day route choice decision simulation based on dynamic feedback information. *Journal of Transportation Systems Engineering and Information Technology*, 10:78–85.
- Unnikrishnan, A. and Waller, S. T. (2009). User equilibrium with recourse. *Networks and Spatial Economics*, 9(4):575–593.
- Wardrop, J. G. (1952). Some theoretical aspects of road traffic research. *Proceedings, Institute of Civil Engineers*, 1(2):325–378.
- Zhong, S. and Ghosh, J. (2001). A new formulation of coupled hidden markov models. Technical report, Tech. Report, Dept. of Electronic and Computer Engineering, U. of Texas at Austin, USA.

# Exploring the Point-mutation Space of a von Neumann Self-reproducer within the *Avida* World

Tomonori Hasegawa<sup>1</sup> and Barry McMullin<sup>1</sup>

<sup>1</sup>The Rince Institute, Dublin City University, Ireland

tomonori.hasegawa2@mail.dcu.ie, barry.mcmullin@dcu.ie

## Abstract

The architecture of machine self-reproduction originally formulated by John von Neumann is studied within the artificial life system *Avida*. We describe a hand-designed von Neumann style self-reproducer, and report initial results from an exhaustive search of its single-point-mutation space. Unsurprisingly, the majority of mutants are simply sterile, and have no long-term evolutionary potential; however, automated characterisation and classification of the minority, fertile mutants proves to be difficult. We identify specific limitations of the standard *Avida* analysis tool for this particular purpose, and outline how it may usefully be enhanced.

## Introduction

The nature of machine self-reproduction was investigated by von Neumann, largely in the early 1950s (von Neumann, 1951, 1966). Inspired partly by Turing's abstract model of computing machines, von Neumann formulated a general architecture for self-reproduction, with a decomposition into active, constructive machinery and a separate, passive "description tape". This work significantly preceded the discovery of the structure of DNA in 1953, but reflects a similar abstract structure to that which is now known to support self-reproduction in biological organisms. Thus, in the von Neumann architecture, the active machinery may be considered as representing the *phenotype* and the passive description tape as the *genotype*.

This von Neumann architecture for self-reproduction is shown schematically in Figure 1. A parent machine (to the left) reproduces an offspring machine (to the right). A self-reproducer in this style consists of a phenotype  $P$  and a genotype  $G$  (or, in an individual, instantiated machine, these may be called "phenome" and "genome" respectively).  $P$  consists of a *programmable constructor* ( $A$ ), a *copier* ( $B$ ), a *control* ( $C$ ), and arbitrary "*ancillary*" machinery ( $D$ ).  $G$  is a tape that describes  $P$  (i.e. the assembly  $A+B+C+D$ ), relative to the specific description language, or "decoding" implemented by  $A$ . In operation,  $A$  decodes  $G$  (to produce another instance of  $P=A+B+C+D$ ),  $B$  constructs a copy of  $G$ , and  $C$  controls and co-ordinates these actions, ultimately detaching the complete offspring machine instance,  $P+G$ ,

identical to the parent and thus realising self-reproduction. As this basic architecture and self-reproducing functionality will be common for any arbitrary  $D$  (within the constructive capabilities of  $A$ , and assuming that  $D$  operations, whatever they may be, do not interfere with  $A+B+C$ ) this implies the existence of an indefinitely large space of self-reproducing machines, all connected via spontaneous perturbations of the  $G$  component (which therefore correspond to heritable mutations). Excluding such perturbation, and in the absence of resource constraints, any specific strain of such machine can exhibit exponential population growth. This potential for exponential growth, combined with mutation (variation) and resource constraints will give rise to conventional, neo-Darwinian, selection and evolution.

A significant additional feature that the von Neumann style of machine self-reproduction can theoretically exhibit is the evolvability of the genotype-phenotype mapping itself (McMullin, 2000) — i.e., of the "decoding" function implemented by the component sub-machine  $A$ . This possibility arises provided  $A$  is itself described (in a self-consistent way) within the genotype,  $G$ , and in sufficient detail that there exist potential mutations (perturbations of  $G$ ) that do change the decoding function implemented by the (expressed, mutated)  $A$  in the following generation. In general, this mutated  $A$  may or may not be capable of decoding the inherited genome (description tape)  $G$  in a way that still preserves the self-reproduction functionality; albeit, the *prima facie* likelihood is that such mutational events will fundamentally disrupt self-reproduction. For a change in the description tape to be truly inheritable, and for a consequent machine to be self-reproducing, the reproduction mechanism must somehow survive through mutational events, sustaining a genotype-phenotype mapping that is still applicable (i.e. "backward compatible") to the mutated description, so as to keep *self*-reproducing. The current paper is concerned precisely with exploring this possibility empirically, at least for one "toy" example of a von Neumann self-reproducer.

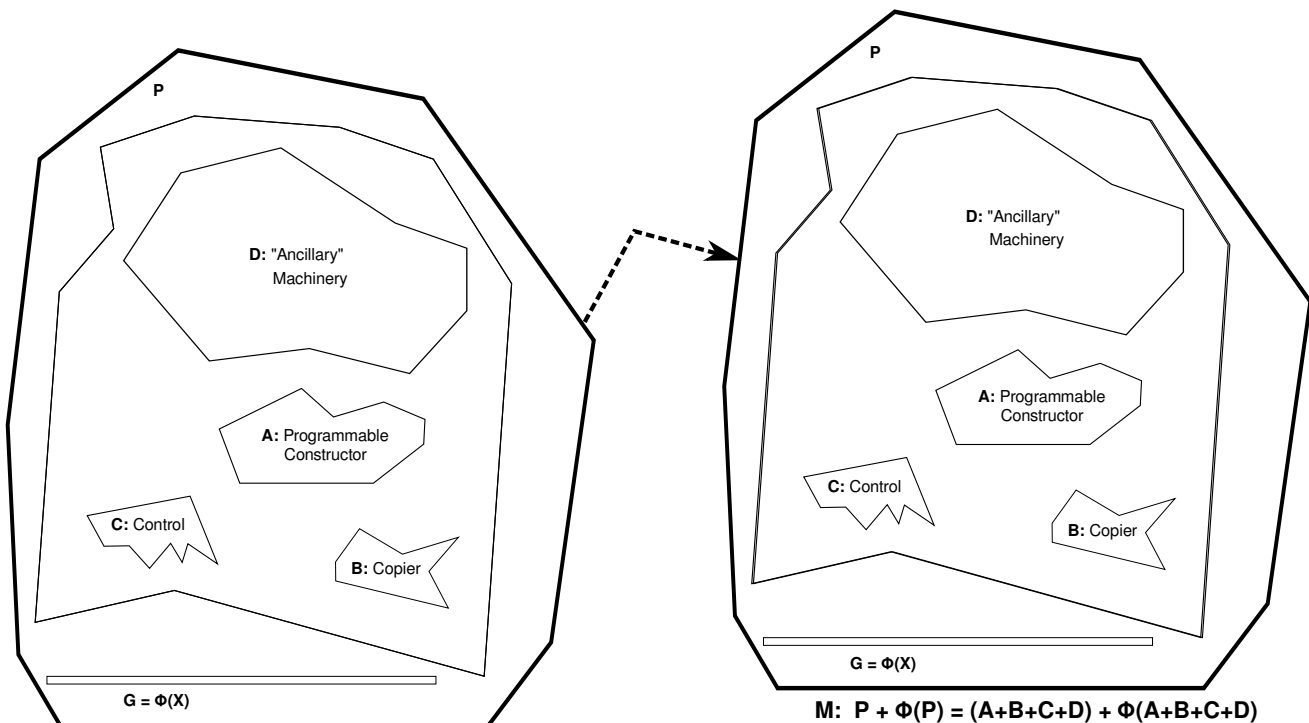


Figure 1: The schematic von Neumann style architecture of machine self-reproduction, excerpted from McMullin (2012).

### The Avida World

Although von Neumann originally elaborated his abstract architecture in the form of cellular automaton works, it can also be successfully implemented within *Core World* type systems. In a typical *Core World* type of system, user-designed machine-code programs (“organisms”) execute, reproduce, and compete for limited resources such as memory space and CPU time. As a result, those systems are expected to display evolutionary dynamics over time. Among others, the *Avida* system (Adami, 1997), which originated in the early 1990s, is an example of such a *Core World* formalism, but with an additional spatial structure. The latter is represented as a homogeneous two-dimensional grid of (virtual) microcontrollers (CPU + small local memory), each of which can instantiate a single running organism.<sup>1</sup> Typically initialised with one seed organism (*ancestor*), one can observe the population growth and evolution in each experimental run.

The standard *Avida* ancestor self-reproduces by means of self-inspection: the program copies its entire memory image word by word to create an offspring’s memory image.

<sup>1</sup>Although the *Avida* world superficially resembles to von Neumann’s early formulation of an abstract cellular automaton (CA) world, there are also fundamental differences. In the von Neumann CA, each node was a simple finite state automaton with no general purpose memory system; whereas each *Avida* node comprises a general purpose CPU and a substantial general purpose memory system.

This latter is (conceptually) divided off, and replaces the memory image in one of the neighbouring cells (see Ofria and Wilke (2004) for more detailed description). Once divided, the parent and the offspring organisms (each with a re-initialised/reset CPU state) continue execution according to their individual configuration (memory images). Such organisms will increase in number, and with mutation, variation may occur among the individuals of the population and give rise to evolution. Our investigation is conceptually similar to this standard *Avida* approach, except that the ancestor organism is designed with a von Neumann architecture instead of as a direct self-copier.

Computationally speaking, a genotype-phenotype mapping in this framework can simply be regarded as an arbitrary, Turing computable mapping between integers (representing the parent and offspring phenotype memory images). In the light of this, a mutable genotype-phenotype mapping through a mutable programmable constructor depends on whether two different such mappings, related via a perturbation on the description tape, can maintain backward compatibility with each other.

We have previously described the implementation and characterisation of a specific prototype von Neumann organism in *Avida*, implementing such a self-reproduction architecture, as a basis for investigating evolvable genotype-phenotype mapping (Hasegawa and McMullin, 2012). This design will now be briefly outlined.

## The Prototype von Neumann Organism

The phenome of the prototype program is coded using possible word contents defined in the Avidan instruction set (see Table 1). Along with the preset standard 26 Avidan instructions, the `read` and `write` instructions are also enabled for this investigation, in order to facilitate flexible reading of the description/genome  $G$ , and writing (construction) of the offspring phenome (memory image). The instruction set configuration in Avida defines what word contents are executable in a particular run, by sequentially associating “op-codes” (mnemonics) with natural numbers according to the order in which they are listed. This set defines possible word contents as numbers from 0 to 27.<sup>2</sup>

For the purposes of this example prototype, the genotype-phenotype mapping (decoding) is chosen to be a simple, sequential, block code. This is inspired by the biological genetic code, relating the sequential primary structures of DNA and corresponding proteins. However, unlike the biological genetic code, where the “alphabets” are different, disjoint, and of different cardinality (nucleotides and amino acids respectively), in our case the alphabets are identical (the distinct values of a single memory location, limited just to the sufficient set to represent each implemented instruction with one op-code, i.e., the numerical range 0..27, per Table 1). Further, choosing a fixed block size of one, this becomes essentially a sequential, monoalphabetic substitution function. This is conveniently implemented via a lookup table, of length 28. Each genome word is sequentially used as an index into this table to find the corresponding “decoded” word to be written into the (offspring) phenome. Note that, with such a coding scheme, a corresponding genome and phenome will always be of equal length; and that, unlike the biological genetic code, there is no redundancy (i.e., each possible phenome word value is represented by one, and only one, possible genome word value).

The designed prototype thus decomposes into the phenome, the first half, and the genome, the second half of the complete memory image. The phenome has five functionally separate regions, namely Decode Preparation, Decode Loop, Copy Preparation, Copy Loop, and Translation Table (see Figure 2 for the prototype’s schematic design and Table 2 for its region allocation and correspondence; find the actual program at our website referenced at the end of the conclusion section). In terms of the generic von Neumann architecture introduced earlier, the Decode Loop along with the Translation Table correspond to the programmable constructor  $A$ , the Copy Loop corresponds to the copier  $B$ , and Decode Preparation and Copy Preparation correspond to the control  $C$ .

The prototype incorporates von Neumann’s architecture

<sup>2</sup>The underlying memory word size is normally 32 bits wide. Numbers beyond the size of the instruction set are associated with the op-codes sequentially in a cyclic manner, so that every possible word value is also interpretable/executable as some instruction.

Word Content	Op-code	Operation
0	<code>nop-A</code>	No-operations
1	<code>nop-B</code>	
2	<code>nop-C</code>	
3	<code>if-n-eq</code>	Flow Control Operations
4	<code>if-less</code>	
5	<code>if-label</code>	
6	<code>mov-head</code>	
7	<code>jmp-head</code>	
8	<code>get-head</code>	
9	<code>set-flow</code>	
10	<code>shift-r</code>	Single Argument Math
11	<code>shift-l</code>	
12	<code>inc</code>	
13	<code>dec</code>	
14	<code>push</code>	
15	<code>pop</code>	
16	<code>swap-stk</code>	
17	<code>swap</code>	
18	<code>add</code>	Double Argument Math
19	<code>sub</code>	
20	<code>nand</code>	
21	<code>h-copy</code>	Biological Operations
22	<code>h-alloc</code>	
23	<code>h-divide</code>	
24	<code>IO</code>	Input/Output and Sensory
25	<code>h-search</code>	
26	<code>read</code>	(Additionally Enabled Operations)
27	<code>write</code>	

Table 1: The instruction set configuration.

except for the (arbitrary) ancillary machinery  $D$ , which is not essential for reproduction per se (i.e.,  $D$  can be null, without violating the abstract architecture) and it is not relevant to our immediate investigation of mutations affecting the genotype-phenotype mapping (i.e. the programmable constructor  $A$ ).

The designed prototype organism has a total memory image of 644 words (i.e. 322 for each half, Phenome and Genome). Note that because the phenome corresponds to the genome on a sequential one-to-one basis, one can locate a unique genotypic word corresponding to any given phenotypic word. In practice, the Phenome segment was designed first, with the Genome segment of the same length being a “black box”. Then, the fully designed Phenome segment was reverse-translated into the corresponding Genome segment, so that the phenome and genome pair can produce the identical pair as an offspring.



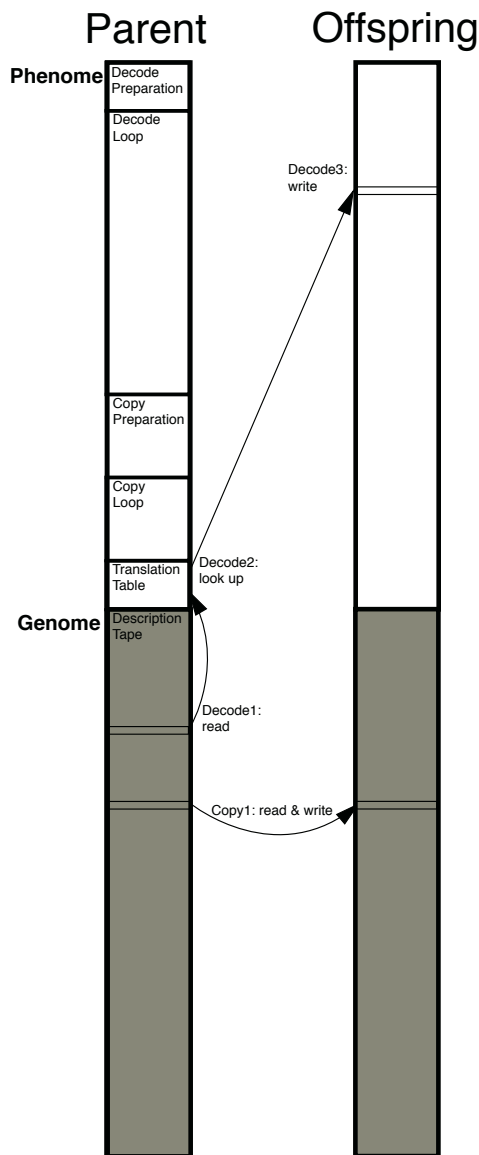


Figure 2: The schematic design of the self-reproduction by the prototype.

Region	Address (in Phenome)	Code Address (in Genome)
<b>Decode Preparation</b>	0–27	322–349
<b>Decode Loop</b>	28–193	350–515
<b>Copy Preparation</b>	194–245	516–567
<b>Copy Loop</b>	246–293	568–615
<b>Translation Table</b>	294–321	616–643

Table 2: The five regions of the prototype’s phenome and the corresponding regions in the genome.

The Translation Table is a “data” segment in the Phenome: it is not to be directly executed (treated as containing instructions), but is referred to, as data, in implementing the decoding. Some concrete substitution code (permutation of the allowed word values 0..27) has to be arbitrarily chosen: we simply used the reversed sequence, 27..0 (i.e., 0 decodes as 27, 1 as 26, etc.).

Once the prototype is seeded in the Avida world, Decode Preparation and Decode Loop are initially executed and decode the genome to create the offspring’s phenome. One step of decoding is as follows: a source word is read from the genome (Decode1), and a destination word is looked up via the Translation Table (Decode2) and is written in a corresponding location in the prospective offspring’s phenome (Decode3). Subsequently, Copy Preparation and Copy Loop are executed and copy the genome to create the offspring’s genome. One word is read and written at one step (Copy1). For a complete self-reproduction, it takes 52218 Avidan instruction cycles (steps). This is the prototype’s *gestation time* (i.e. reciprocal of reproduction rate).

In previous work, we observed the prototype’s behaviour and characteristics as an ancestor. The design of the prototype proved to work out correctly, demonstrating one implementable instance of a von Neumann style self-reproducer within Avida, a Core World type of system. A noticeable finding from this work was that the self-reproducer can degenerate into a pure self-copier only through one step of single-point mutation in the course of evolution. That is, there is at least one possible mutational pathway for the ancestor to become a self-copier.<sup>3</sup> Though this could presumably happen through a longer gradual evolutionary process, the finding in this context suggests that even one step of

<sup>3</sup>Thus, it is theoretically possible to vice versa also: that the prototype can degenerate into a self-copier means that it is likewise possible for a self-copier (or, at least, that particular self-copier) to evolve to a von Neumann self-reproducer.

single-point mutation can bring about unpredictable changes in the behaviour of the prototype.

Still, it is not clear yet how typically the prototype potentially degenerates (or exhibits other similar phenomena); therefore it is legitimate to focus on the particular von Neumann style self-reproducer and to extend the investigation of its mutational pathways. This study can reasonably be a springboard to better understand the von Neumann style self-reproduction within the system, considering the vast space of possible strains of organisms in the current Avida setting. The space appears to roughly consist of reproducers and non-reproducers; further, the reproducers include self-reproducers and other kinds of reproducers; furthermore, self-reproducers may comprise self-copiers, von Neumann style self-reproducers and some other classes.

### Empirical Investigation: Point-mutation Space Search

For a better picture of the mutational pathways originating from the prototype, we first systematically counted out possible single-point mutants as candidates for organisms that have evolutionary potential, especially ones that maintain von Neumann style self-reproduction.

Candidates are obtained from the prototype's initial memory image by sequentially replacing a word content in each memory location of the genome with the other possible word contents listed in the instruction set and by expressing this change in a corresponding phenome. Each such candidate is an organism in Avida; or more properly, it is an initial memory image coupled with an initial virtual CPU configuration. Now, the genome size is 322 (the half of the whole length of the prototype), and the size of the current instruction set is 28. Considering the combination of the genotypic memory locations and the different possible word contents for each memory location (i.e., excluding the one originally existing in the location), the number of candidates is therefore 8694 ( $= 322 \times (28 - 1)$ ).

For the purpose of characterising the prototype, the candidates should ideally be classified by the "mode" of reproduction. However, reproduction mode classification is not straightforward, since a reproduction mode is not simply determined by a single attribute of reproduction or lineages, measurable when an organism is run and traced. Nonetheless, there are at least two distinct self-reproduction modes: the pure self-copying by inspection as in the standard Avida ancestor and the von Neumann style as in the prototype. Also, one can speculate that there may be various modes aside from, or in between, these two. At any rate, to effectively judge the reproduction mode, it is generally useful to analyse in combination the candidates' attributes (e.g. gestation time) and their execution profiles (e.g., instruction pointer traces and execution counts of particular instructions, such as the `write` and `h-copy` instructions, which write some content in a memory location). They, however,

do not yet necessarily guarantee what reproduction mode that a candidate uses. Rather, we need to observe how each offspring is created; besides, *viability* cannot be determined until the candidate's lineage is wholly traced for a proper number of generations.

As a preliminary step of the investigation, we attempted to classify the candidates by viability. Viability was judged initially by the Avida built-in analysis tool (specifically, using a `TRACE` command in a mode called *analyze mode*), which can trace a single lineage pathway starting from an incubated organism.

### The Analyze Mode

Technically, according to the `TRACE` command in the Avida *analyze mode*, an organism running for a given generation is classified into one of four possible classes:

- If no division occurs within a predefined cut-off time, the mode concludes that the starting organism is *non-dividing* (meaning non-reproducing); or else,
- If division occurs and the immediate offspring is identical, the starting organism is classified as *immediate self-reproducing*; or else,
- If division occurs and the offspring is not identical to the immediate parent, but identical to any of the (already traced) ancestors, then the starting organism is classified as *indirect self-reproducing*; or else,
- If division occurs and the offspring is identical to none of the direct ancestors, the starting organism is classified as *reproducing* (meaning non-self-reproducing); this final class potentially triggers a recursive analysis one further generation deep, unless terminated by reaching a maximum depth.

This automated analysis assumes a cutoff time, a certain window of maximum runtime, so as to judge as non-dividing or not. In order to be classified as some reproducing class, division must have taken place (i.e., the `h-divide` instruction must have been executed) before the cutoff time is reached; otherwise, the analysis run terminates classifying the organism as non-dividing. The cutoff time is set twice as long as the prototype's gestation time (i.e.  $104436 = 2 \times 52218$ ). This setting is reasonable considering the fact that the candidates can have varied gestation times. It is possible that, with a longer runtime, some candidates which are classified as non-dividing might be reclassified as reproducing. They are, however, unlikely to be selectively favoured over the original prototype ancestor in the Avida system, where fitness hinges on reproduction rate hence gestation time.<sup>4</sup> On the grounds of this, we can heuristically discard those classified as non-dividing.

<sup>4</sup>As opposed to these, the ones with shorter gestation time can potentially outnumber the others in the world where it boils down to the fitness, especially when the run is sufficiently long.

Out of the four classes, the non-dividing class indicates non-viability whereas the immediate self-reproducing class and the indirect self-reproducing class indicate viability. On the other hand, the reproducing class does not necessarily indicate viability or non-viability, as the offspring may or may not be self-reproducing; therefore only this class requires more generations to track down the lineage pathway in order to determine the viability. The analyze mode applies this classification repeatedly over generations as necessary to judge viability.

It is important to note that, in fact, the analyze mode only traces the offspring that is divided off from the parent at each division, assuming that viability means the ability to sustain self-reproduction on a single lineage pathway. To clarify this point, consider that each division can be regarded as making two offspring: one that used to be a parent, the other that has been divided off. The former replaces the parent and the latter replaces one cell in the neighbourhood, so spatially it appears that the parent remains sitting in the original cell and producing and placing one offspring into an adjacent cell. What is traced by this tool is the *divided-off* offspring, not the *sitting* one: only a single lineage pathway is revealed for each incubated organism (as opposed to the whole lineage with possibly multiple pathways).

### First-division Patterns

To support the somewhat limited analysis by the built-in tool, we applied a more refined classification based on certain patterns characteristic of the first-division event (for all cases other than simple *non-dividing*). This reclassification helps clarify the lineage pathways one-step further.<sup>5</sup> As pointed out earlier, division makes two organisms: the two offspring that the initially placed organism produces. One of the two offspring should be labelled as the organism that used to be the parent, and the other offspring as the organism divided off from the parent, which treatment is somewhat arbitrary. There are patterns in these three memories, depending on which of them are the same and/or different. Here, assume that an organism has an initial memory ( $I$ ; the parent's initial memory image) which becomes a final memory ( $F$ ; one of the offspring's initial memory image) when dividing off a child memory ( $C$ ; the other offspring's initial memory image). There are six division patterns at one attempt of division. Assume they are notated as follows using  $I$ ,  $F$ , and  $C$  for the sake of convenience:

- $I$ : Non-dividing;
- $I = F = C$ : Self-reproducing. Both of the offspring's initial memory images are identical to the parent's initial memory image;

<sup>5</sup>Although ideally all the attempts of division should be classified to investigate each organism's whole lineage, the automation of such classification requires considerable enhancement as discussed later.

- $I = F \neq C$ : One of the offspring's initial memory image is identical to the parent's, while the other offspring's is non-identical;
- $I = C \neq F$ : Essentially the same as the previous one, but this pattern is treated as self-reproducing by the analyze mode;
- $I \neq F = C$ : The two offspring's initial memory images are non-identical to the parent's, but identical to each other; and
- $I \neq F \neq C$ : Neither of the two offspring's initial memory images is identical to the parent's or to each other.

The first attempts of division in the candidates' single lineage pathways revealed by the analyze mode were studied and the candidates were reclassified based on these patterns.

## Results

The whole candidates distribute as shown in Table 3 when classified by "viability" using the built-in analysis tool and when by "fertility" based on the first-division patterns. As the analyze mode's ability to trace lineage pathways is not full, its "viability" judgment is different from what we originally meant: by viability we originally meant some continuous reproducibility with evolutionary potential. Here, in order to describe what is revealed by the analysis, we define *fertility* as follows: an organism is classified as fertile if its lineage is fully traced and in the lineage there is at least one strain that is classified as immediate or indirect self-reproducing.

Initially, of the whole candidates, the immediate self-reproducers (fertile, but only judged by tracing one offspring) accounted for nearly 10%, the pure non-reproducers (fertile, being non-dividing) for nearly 60%, and the rest 30% (fertile, but at least not immediate self-reproducing) remained unclassified.<sup>6</sup> The reclassification based on the first-division patterns revealed some of the unclassified that are infertile and discovered the others that have an untraced lineage pathway(s).

In other words, out of those initially classified as "viable", some lineages turn out not to have been fully traced; even though it is shown by the analysis tool that they have at least one viable reproduction pathway, they should be reclassified as unclassified. Those initially classified as "non-viable" are wholly reclassified as infertile without further scrutiny as they have no division executed. Out of those initially unclassified, some lineages turn out to have been fully traced, whether it be fertile or infertile, now putting fewer candidates in the unclassified.

<sup>6</sup>In hindsight, the analyze mode found no case of the indirect self-reproducing class among the candidates. As for the reproducing class, it found a few long cases with more than 300 recursions.

Immediate Classification	“Viable”	Unclassified	“Non-viable”	Total
Total	892 (10%)	2588 (30%)	5214 (60%)	8694

1st Division Pattern	Fertile	Unclassified	Infertile	Total
I	0 (0%)	0 (0%)	5214 (100%)	5214
$I = F = C$	871 (100%)	0 (0%)	0 (0%)	871
$I = F \neq C$	0 (0%)	158 (8%)	1745 (92%)	1903
$I = C \neq F$	0 (0%)	21 (100%)	0 (0%)	21
$I \neq F = C$	1 (3%)	11 (28%)	27 (69%)	39
$I \neq F \neq C$	0 (0%)	646 (100%)	0 (0%)	646
Total	872 (10%)	836 (10%)	6986 (80%)	8694

Table 3: Automated mutant “viability” classification (Top) and mutant fertility classification with the first-division patterns (Bottom).

Consequently, the fertile account for 10%, the infertile for 80%, and the unclassified (fertile but not immediate self-reproducing; may be infertile, reproducing, or indirect self-reproducing) for 10%. The combination of classifications allowed us to clarify which lineages of the candidates have been adequately traced and how much we have known about their viability.

## Discussion

The whole candidates were classified using the built-in analysis tool and based on the first-division patterns. Viability classification was not full lineage analysis due to some technical limitation. Nevertheless, viability was revealed to the extent that fertility classification revealed infertile, hence non-viable, candidates. The still unclassified candidates are all fertile but have untraced lineage pathway(s). With full lineage analysis, those candidates can be reclassified either as fertile (i.e., having some self-reproducing strain in lineage) or as infertile (i.e., having no self-reproducing strain

in lineage). From there, we intend to distinguish viability of the candidates, which indicates not only fertility but also evolutionary potential. These unclassified candidates are not negligible since we do not know a priori how frequently the candidates are fertile or viable. The classification should ideally be automated and systematised, even applicable to different sets of candidates.

## Search Limitation and Possible Enhancement

To emphasise the problem situation, the limitation of the current analysis tool surfaces when it hits either cases where (a) both of the two offspring that a parent produces are different from the parent or from each other, or where (b) one of the two offspring is the same as the parent but the other is different.<sup>7</sup> This fact implies that the built-in analysis tool assumes that even though an act of division produces two same offspring, tracing one of them suffices to analyse the lineage. Again, when the built-in analysis tool searches for descendant generations further down, what it recursively incubates for tracking is only one of the two offspring that is arbitrarily labelled as being divided off from the parent.

For example, suppose an organism produces one non-dividing offspring and one reproducing offspring in a deterministic environment without mutation. If the reproducing offspring produces a self-reproducing offspring, the original organism should be classified as (indirect) self-reproducing in that it exhibits the lineage. However, if the analysis tool is set to trace one of the offspring at the first-division, which happens to be non-dividing, then the other offspring, which happens to be reproducing a self-reproducing offspring, is not further traced, and the original organism ends up being classified as “non-viable”. This is not a proper viability classification because self-reproducers may be viable, with an evolutionary potential.

In other words, for some candidates which start out producing different two offspring, only a subset of (or rather, a “branch” of) the whole expected lineage becomes revealed through the analysis tool. Therefore, we can neither guarantee their fertility nor viability. Fertility implies reproducibility, whereas viability implies fertility with evolutionary potential, something that, for example, leads to an exponential growth of population (e.g. as pure self-reproducers), as opposed to a linear population growth<sup>8</sup>. Importantly, not only individually self-reproducing strains but also “collectively” self-reproducing strains (i.e., strains that are self-reproducing from a lineage point of view) should be classified as fertile, and even as viable, being evolutionarily po-

<sup>7</sup>The case (a) applies to the pattern  $I \neq F \neq C$  and the case (b) applies to the patterns  $I = F \neq C$ ,  $I = C \neq F$ , and  $I \neq F = C$ , in the notation introduced earlier.

<sup>8</sup>The *pathological constructor* proposed by Baugh and McMullin (2012) is an example, the strain of which reproduces itself and an infertile strain. This would exhibit the pattern  $I = F \neq C$ .



tential.<sup>9</sup> All this renders the classification non-trivial.

The automated analysis proceeds with the time limit as mentioned earlier and implicitly with the recursion limit (i.e., how many generations to track down deeper). Aside from these limits, because of the lineage traceability limit discussed above, not all of the candidates have been analysed for viability, although the majority of the candidates have been analysed for fertility.

It is our next, ongoing step to modify the analysis tool with an enhanced ability to fully cover the whole lineage pathways expected from an organism (although of course practically there should be set a division time limit and a recursion limit in automatically revealing lineages).

## Conclusions

In the previous research, we constructed and observed a von Neumann style ancestral self-reproducer with a genotype-phenotype mapping subject to evolution within the particular platform. The discovery of the self-reproducer's quick evolutionary degeneration into a self-copier raised a question: How likely is it such degeneration takes place? To answer this question, in the present research, we endeavoured to investigate the spectrum of single-point mutants of the particular self-reproducer in an attempt to classify *viable* candidates for evolvable genotype-phenotype mapping. The presented results are rather preliminary and should serve as a platform to further characterise the prototype self-reproducing in a von Neumann style. In the course, we consequently encountered a new situation where we ask: Given spectrum of mutants, can we classify them by viability?

The automated analysis, nevertheless, yielded an insight as to distinguishable *fertile* candidates, when combined with the classification of the first-division patterns. The fertile candidates (10%) need further scrutiny for reproduction mode and evolutionary potential, while the infertile candidates (80%) need not since they are logically non-viable. At any rate, while the vast majority (90%) of the whole candidates are classified as either fertile or infertile, the rest (10%) of the whole candidates yet remain unclassified.

The existence of those still unclassified candidates clarified the fact that there is a subtlety in the concept of viability. Specifically, the current analysis tool is only capable of revealing a single possible lineage pathway that an organism is expected to exhibit. To recapitulate, it traces only one (arbitrarily pre-determined) of the two offspring at each division, whereas the sub-lineage extending from the other offspring is not further traced. In this sense the analysis is not thorough. It is therefore necessary to automatise and systematise the classification of candidates based on viability, and further, based on the mode of self-reproduction, in the current system. The built-in analysis tool is being enhanced to

thoroughly investigate lineage pathways. A better characterisation of the particular example of a von Neumann style self-reproducer requires to understand its mutational pathways from a viewpoint of viability.<sup>10</sup>

The experimental set of the Avida system presented in the current paper, including the ancestral program file and the instruction set configuration file, can be accessed at: [http://alife.rince.ie/evosym/ecal\\_2013\\_thbm.zip](http://alife.rince.ie/evosym/ecal_2013_thbm.zip).

## Acknowledgements

This work has been supported by the European Complexity Network (Complexity-NET) through the Irish Research Council for Science and Technology (IRCSET) under the collaborative project EvoSym. We would like to appreciate the original developers of the Avida platform. Furthermore we would also like to acknowledge the three anonymous reviewers for their valuable comments and suggestions to improve the consistency of this paper.

## References

- Adami, C. (1997). *Introduction to Artificial Life*. Springer, corrected edition.
- Baugh, D. and McMullin, B. (2012). The emergence of pathological constructors when implementing the von neumann architecture for self-reproduction in tierra. In Ziemke, T., Balkenius, C., and Hallam, J., editors, *From Animals to Animats 12*, pages 240–248, Berlin. Springer.
- Hasegawa, T. and McMullin, B. (2012). Degeneration of a von neumann self-reproducer into a self-copier within the avida world. In Ziemke, T., Balkenius, C., and Hallam, J., editors, *From Animals to Animats 12*, pages 230–239, Berlin. Springer.
- McMullin, B. (2000). John von neumann and the evolutionary growth of complexity: Looking backward, looking forward. *Artificial Life*, 6(4):347–361.
- McMullin, B. (2012). Architectures for self-reproduction: Abstractions, realisations and a research program. In Adami, C., Bryson, D. M., Ofria, C., and Pennock, R. T., editors, *Artificial Life 13*, pages 83–90. MIT Press.
- Ofria, C. and Wilke, C. O. (2004). Avida: A software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229.
- von Neumann, J. (1951). The general and logical theory of automata. *Cerebral mechanisms in behavior*, pages 1–41.
- von Neumann, J. (1966). *Theory of self-reproducing automata*. University of Illinois Press. Edited and completed by A.W. Burks.

<sup>9</sup>This distinction appears to be in resonance with that between individual autocatalysis and collective catalysis in artificial chemistry.

<sup>10</sup>Furthermore, to effectively find interesting reproduction modes in viable candidates, it may be practical to reclassify the candidates based on the likelihood of experiencing a change in genotype-phenotype mapping. In the design of the prototype, the regions where mutation most likely can cause any change in genotype-phenotype mapping are Translation Table and Decode Loop, both serving as a part of the programmable constructor *A* of von Neumann's architecture.

# Emergence of diverse behaviors from interactions between nonlinear oscillator complex networks and a musculoskeletal system

Hiroki Mori<sup>1</sup>, Yuzi Okuyama<sup>1</sup> and Minoru Asada<sup>1</sup>

<sup>1</sup> Department of Adaptive Machine Systems, Graduate School of Engineering, Osaka University, Japan,  
hiroki@sms.eng.osaka-u.ac.jp

## Abstract

To understand the relationship between brain structure and behavior in the general movements of fetuses and infants from a complex systems perspective, we investigated how behaviors emerge from interactions between complex networks of nonlinear oscillators and musculoskeletal bodies. We prepared a snake-like robot and some network structures in a physical simulator. The various conditions imposed on the networks were (a) no connection among oscillators, (b) scale-free network, (c) one-dimensional lattice, (d) small-world network, and (e) random network. In the experiments, the robot exhibited multiple crawling and bending behaviors. By estimating the numbers of behavioral attractors, we revealed a qualitative difference between the scale-free network and other complex networks.

## Introduction

Animals and humans exhibit various adaptive behaviors. These behaviors likely emerge from complex interactions among environment, body dynamics, and brain activities, rather than from any single factor, posing several questions. How do the behaviors emerge? What underlying structure shapes such a complex and adaptive interaction? Do diverse interactive behaviors among systems emerge under any specific condition? To answer these questions, we must understand the relationship between structure and functionality in biological entities from a complex systems perspective.

Kuniyoshi and Suzuki (2004) proposed a model in which adaptive behaviors emerge through body constraint as a chaotic coupled field. In this model, on the basis of coupled map lattices and globally coupled maps (Kaneko and Tsuda (2003)), transitional adaptive behaviors should emerge as chaotic itinerancy among behavioral attractors. The model connects each chaotic element (logistic map) to a single linear actuator (muscle); that is, the actuator receives the chaotic elements output as a motor command and sends the outputs of its length sensors back to the chaotic element. The body structure behaves as a coupled field of deterministic chaotic activities. However, when the model is executed, the number of emerged behaviors is less than expected, indicating that chaotic itinerancy was suppressed.

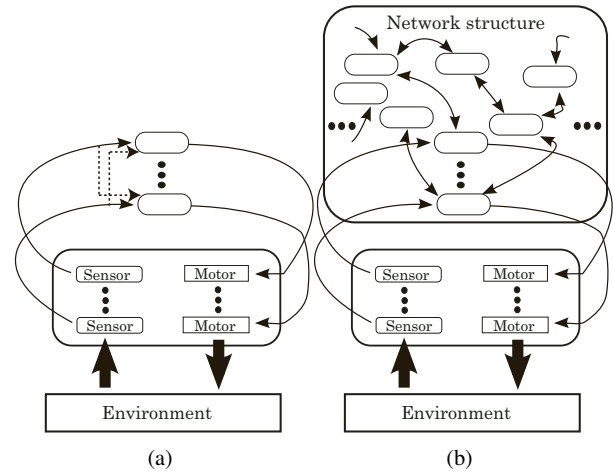


Figure 1: Emergence of behaviors modeled as interactions among nonlinear oscillators and bodies. (a) The model of behavioral emergence based on chaotic coupled field (Kuniyoshi and Suzuki model). (b) Model proposed in this paper. The emergence model interacts with nonlinear oscillators possessing a certain complex network structure.

Complex network structures in the human brain are regarded as either small-world networks (Watts and Strogatz (1998)) or Scale-free networks (Barabasi and Albert (1999)). We are interested in the functional role of such brain structures.

This paper proposes a first step for investigating the emergence of organisms versatile behaviors by a constructive approach. To this end, we constructed a simulator built from complex nonlinear oscillator networks and a snake-like musculoskeletal body model. We analyzed and compared the emergent behaviors from the coupled dynamics of the musculoskeletal body and complex network structures.

## Behaviors and brain structures

### General movements of fetuses and infants

General movements, defined as purposeless whole-body movements, characterize the movements of fetuses and infants. The movements begin at about eight weeks gestation, and continue to evolve in healthy infants after birth. Within the first two months of birth, infants develop writhing movements, characterized by small-to-moderate amplitude and slow-to-moderate speed Prechtl (2001), although the arms may move rapidly and largely at this stage. After two months, fidgety movements appear, which are characterized by simultaneous variable acceleration of all moving parts of the body with other gross movements. After about 20 weeks, infants gradually develop voluntary movements.

Such movements are considered to be generated by the medulla oblongata in the brain stem. Anencephalic infants, who lack the neocortex, can move their whole body roughly and rapidly. Additionally, since the quality of general movements is related to white matter, the neocortex might be responsible for movement control Spittle et al. (2008).

### Brain structure and complex network theory

What kind of structure exists in the brain? This question has been answered by diffusion tensor imaging (DTI), which visualizes the nervous fiber connectivity in white matter. The brain structure is a variant of small-world networks, known as a rich-club network (van den Hauvel and Sporns (2011)).

A small-world network is one in which any two nodes are connected through very few interceding nodes, compared to the network size. Mathematically, in a small-world network, the average minimum distance  $d_{ji}$  between two nodes  $i$  and  $j$  is proportional to the logarithm of the total number of nodes  $N$  in the network. Thus, the small-world network satisfies (Watts and Strogatz (1998))

$$d \propto \log(n). \quad (1)$$

Scale-free networks are frequently encountered in complex network theory. In this type of network, connections are concentrated among several nodes, and the majority of nodes have only a few connections. Scale-free networks are characterized by the power law

$$P(k) \propto k^{-\gamma}, \quad (2)$$

where  $k$  is the number of edges from a node,  $P(k)$  is the probability of  $k$ , and  $\gamma$  is a constant. These networks are considered to be controlled by a small number of hub nodes.

### Brain rhythms and nonlinear oscillators

Brain activities manifest as brain waves of certain frequencies, which are synchronized or desynchronized depending on internal and external situations (Buzsaki and Watson (2012)). This synchronicity is called functional connectivity, while the anatomy of the nervous fiber tracts observed by

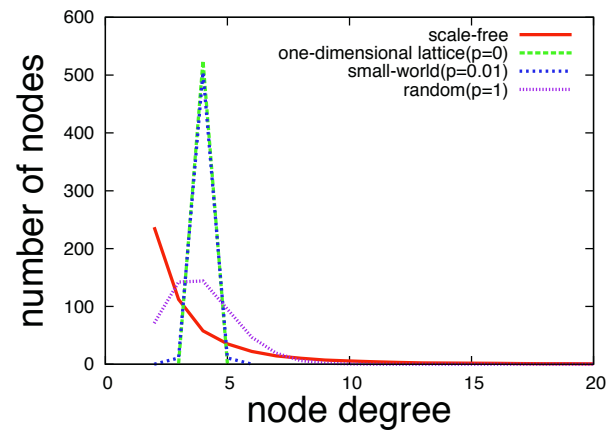


Figure 2: Distribution of degree of connection to one node in each network

DTI is called structural connectivity (Sporns (2012)). One of our interests is to elucidate how anatomical structures induce functional structures.

Oscillatory activities are observed not only in the neocortex but also in the brain stem. As mentioned above, such oscillatory activities are often modeled by nonlinear oscillator equations. As is well known, coupled nonlinear oscillators show deterministic chaotic activities such as synchronization, desynchronization, and great chaos, which is an exploratory behavior (Tsuda et al. (2004); Asai et al. (2000)).

### Chaotic itinerancy by chaotic activities coupled through a body structure

Kaneko and Tsuda (2003) proposed two coupled chaos models – coupled map lattice (CML) and globally coupled map (GCM) – which are suitable for general investigation of complex systems. They found that the models generate both ordered and disordered patterns such as deterministic chaotic itinerancy. Inspired by these models, Kuniyoshi and Suzuki (2004) proposed that adaptive and exploratory behaviors such as chaotic itinerancy could be induced by coupling chaotic elements through a robotic body as a physical constraint. As mentioned above, this model shows adaptive behaviors in varying environments, but exploratory behaviors are fewer than expected.

### A model based on interactions between the body and complex networks.

#### Complex network

Several real-world networks are considered to be scale-free or small-world. Such a network structure applied to chaotic elements generates synchronized clusters among chaotic elements (Jalan et al. (2005)).

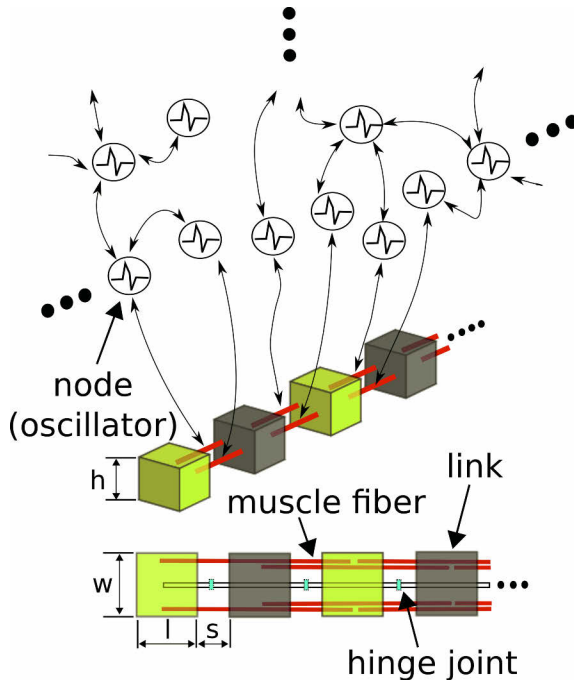


Figure 3: Conceptual diagram of the model

**Small-world network and random network** Watts and Strogatz (1998) proposed that small-world networks can be generated by reconnecting a coupled map lattice as follows:

1. Start from a ring lattice of  $n$  nodes coupled to neighboring nodes up to a specified distance  $m$ .
2. Rewire each edge randomly with probability  $p$ .

z The behaviors of these networks are varied by reconnecting each connection in CML with probability  $p$ . The network is purely CML and completely random at  $p = 0$  and  $p = 1$ , respectively. Typically, small-world properties emerge between  $p = 0.01$  and  $p = 1$ .

**Scale-free network** In this study, we implemented a scale-free network using the following algorithm, known as the Barabasi-Albert (BA) model (Barabasi and Albert (1999)).

1. The algorithm begins with a single node.
2. Count  $k_i$ , where  $k_i$  represents the degree of node  $i$
3. Calculate  $P(k_i)$  by dividing  $k_i$  of each node by the sum of the degree of all nodes (Equation (3)).

$$P(k_i) = \frac{k_i + 1}{\sum_j (k_j + 1)} \quad (3)$$

4. Add a new node with  $m$  connections to existing nodes, on the basis of the probability of coupling  $P(k_i)$ .

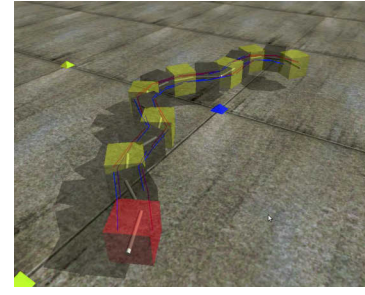


Figure 4: Appearance of our snake-like robot with the muscles

5. Repeat steps 2-4 until a specified number of nodes is reached.

In the BA model, the number of connections to a single node in the network obeys the power law. Many real-world networks, such as the world wide web and human societal relationships, are known to be scale-free, and have been actively researched in recent years. Scale-free networks are also a type of small-world networks because of their structural characteristics. We have confirmed that the BA algorithm yields short average distance between two nodes, relative to the network size. Therefore, in a broad sense, we regard scale-free networks as small-world.

### Nonlinear oscillators

Our model adopts two types of nodes (nonlinear oscillators): output nodes, which directly connect to the body, and hidden nodes, which connect to each other with no direct connection to the body. Output nodes activate muscle fibers and receive feedback from length sensors of muscles, while hidden nodes affect the behavior of the model through the network. Each oscillator is represented as a boundary value problem (BVP) (Equation (??)) that behaves like the action potential of neurons. Multiple coupled BVPs are known to bifurcate (Equation (6)) and consequently display complex behaviors (Asai et al. (2000)).

There are two types of nodes (nonlinear oscillators) in our model. One of the types are output nodes, which have connections to the body directly, and another type are hidden nodes, which connected each other and have no direct connection to the body. Output nodes activate muscle fibers and receive feedback from length sensors. Hidden nodes affect the behavior of the model through the network. Each oscillator is represented by a BVP equation ((Equation (4)) and (Equation (5))) that behaves like the action potential of neurons. It is known that bifurcation occurs in multiple coupled BVPs (Equation (6)), which induce complex behaviors (Asai et al. (2000)).

$$\frac{dx_1[i]}{dt} = c(x_1[i] - \frac{x_1[i]^3}{3} - y_1[i] + z) + \delta(S_{other}[i] - x_1[i]) \quad (4)$$



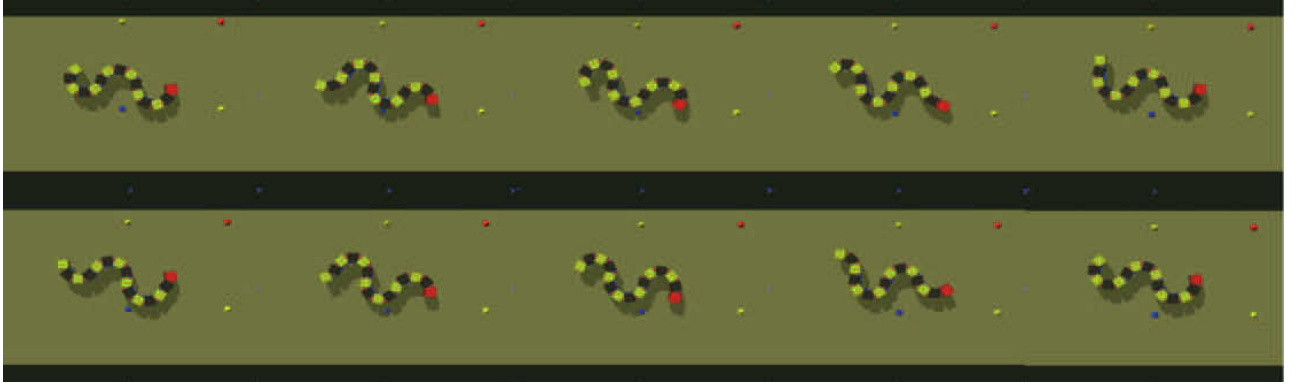


Figure 5: Example of crawling movement by our snake-like body. Time series runs from right to left, and from top to bottom.

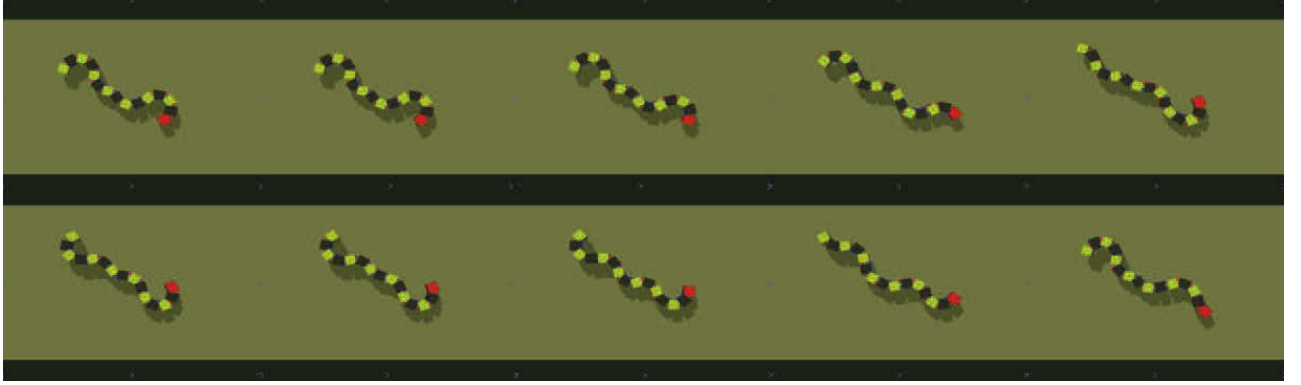


Figure 6: An example of bending movement by our snake-like body. Time series runs from right to left, and from top to bottom.

$$\frac{dy_1[i]}{dt} = \frac{1}{c}(x_1[i] - by_1[i] + a) + \epsilon S_{other}[i] \quad (5)$$

$$S_{other}[i] = \begin{cases} \sum_{j=1}^n \frac{C_{i,j}x_1[j]}{k[i]} & (HiddenNode) \\ \frac{1}{2}(gs_{in}[i] + \sum_{j=1}^n \frac{C_{i,j}x_1[j]}{k[i]}) & (OutputNode) \end{cases} \quad (6)$$

In the above expressions,  $x_i[i]$  and  $y_i[i]$  denote the action potential and inhibitory potential of node  $i$ , respectively, and  $z$  is the tonic input.  $S_{in}[i]$  is the sensor feedback from the muscle fibers, while  $g$  is the gain of sensor values.  $a, b, c, \epsilon, \delta$  are constants.  $k[i]$  is the degree of node  $i$ .  $C_{ij}$  is a weight matrix that represents the coupling state in the network. The tonic input controls the stability of oscillations; the higher the tonic input, the more chaotic the oscillators. The sensory gain controls the strength of connection between the musculoskeletal and oscillator systems.

### Network structures of nonlinear oscillators

Experiments were conducted on the following nonlinear oscillator networks.

**Experiment (a)** No connections among the nonlinear oscillators. Each oscillator connects to its corresponding muscle.

**Experiment (b.1)** Scale-free network, modeled by BA with no hidden nodes.

**Experiment (c.1)** One-dimensional lattice. Nonlinear oscillator network, modeled by the algorithm of Watts et al. with  $p = 0$ .

**Experiment (d.1)** Small-world network. Nonlinear oscillator network, modeled by the algorithm of Watts et al. with  $p = 0.01$ .

**Experiment (e.1)** Random network with no hidden nodes. Nonlinear oscillator network, modeled by the algorithm of Watts et al. with  $p = 1$ .

**Experiment (b.2)** Scale-free network, modeled by BA with 500 hidden nodes.

**Experiment (c.2)** One-dimensional lattice with 500 hidden nodes

**Experiment (d.2)** Small-world network with 500 hidden nodes.

**Experiment (e.2)** Random network with 500 hidden nodes.

To examine the effect of network size on the emergence of behaviors, we conducted experiments without hidden nodes and with 500 hidden nodes on all network models. Each connection weight was

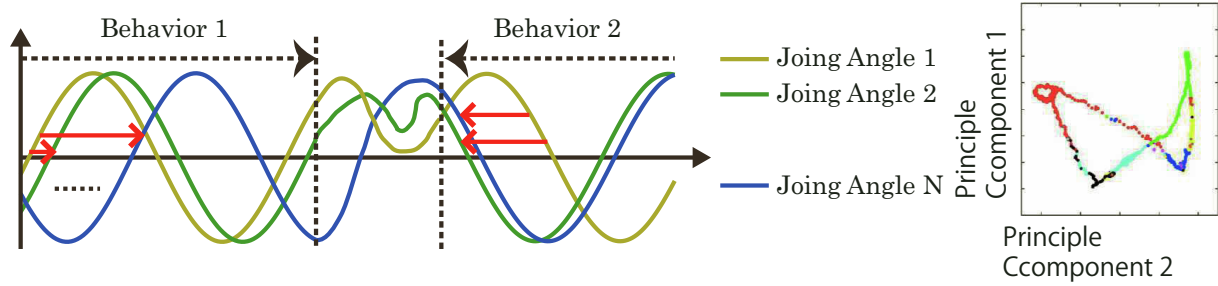


Figure 7: Left: Conceptual diagram of the joint angle data at the time of behavioral transition. Right: An example of clustering of a trajectory in two principle components of the phase space, obtained by mean-shift clustering method. Each color represents a cluster. Finally, trivial clusters involving less than 10 % of the data points are eliminated from the estimated behavioral attractors.

Table 1: Average shortest distance between two nodes in each network with 500 hidden nodes

scale-free	one-dimensional lattice $p=0$	small-world $p=0.01$	random $p=1$
3.98	66.125	22.85	4.79

Table 2: Specifications of snake-like body

link of height	link of width	link of length	gap of two bodies	Mass of a link
0.1 [m]	0.1 [m]	0.1 [m]	0.02 [m]	0.6 [kg]
Number of links		Number of one joint muscles		Number of two joint muscles
15		0		26

randomly drawn from a uniform random distribution. The connection weights to each node were normalized to a norm of 1. Although the connections between nodes were bidirectional, the connection weights were determined independently for each direction. The degree distribution in networks with 500 hidden nodes is displayed in Figure 2. Table 1 shows the average distance between nodes in the networks.

### Snake-like body

The muscles of the snake-like body in our model are polyarticular, enabling more synchronous body movements (Niiyama et al. (2007)). Table 2 and 3 provide an overview of the snake-like body.

Simulations were performed using Open Dynamics Engine (ODE) Smith (2001). Our model consists of a set of nonlinear oscillators and the snake-like body. The nonlinear oscillators are separated into two categories: One comprises output nodes, which generate motor commands sent to muscles and receive the length sensor outputs from the muscles. Another comprises hidden nodes, which are not directly connected to the muscles.

## Experiments and Analyses

### Experimental settings

Experiments were conducted on the abovementioned networks. To examine the effects of the system parameters, we varied the tonic input and sensor gain parameter. For each parameter set, 100 experiments were performed with different randomized weights.

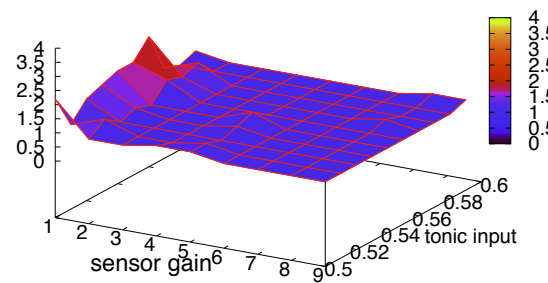


Figure 8: Number of behavioral attractors without any network structures

### Number of periodic motion patterns of the body

We estimated the number of periodic behaviors in the system by analyzing the time series of the robots joint angles, as described below.

1. Fourier-transform the series of joint angles over a shifting time window. In this study, the width of the time window is 5 [sec], shifting with a time step of 0.1 [sec]. After Fourier transformation, the frequency of maximum amplitude identifies the main component of the oscillatory behaviors in each joint. Accordingly, we refer to this wave of maximum amplitude as the representative wave.
2. Calculate the phases of the representative waves from the reference joint angle (Figure 7). Each point in the multidimensional phase space represents a single periodic movement.
3. For clustering, reduce the number of dimensions by principal component analysis. We selected five principle components, thereby retaining 90 % of the variance.
4. Estimate the number of periodic behaviors by applying mean-shift clustering (Comaniciu and Meer (2002)) to the data analyzed by principle component analysis.
5. Count the clusters. To reduce trivial behaviors, omit clusters containing fewer than 10 % of the data points.

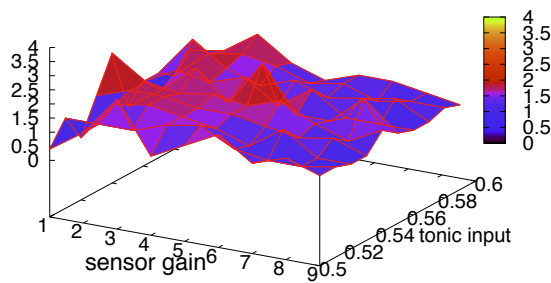


Figure 9: Number of behavioral attractors with a scale-free network (hidden node:0)

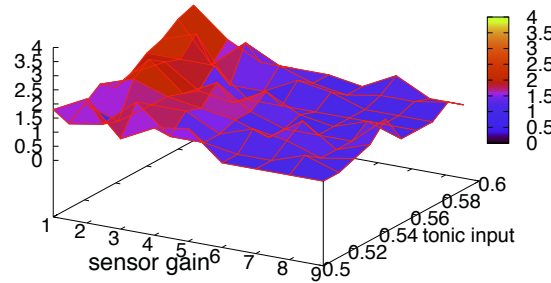


Figure 11: Number of behavioral attractors with a small-world network (hidden nodes = 0)

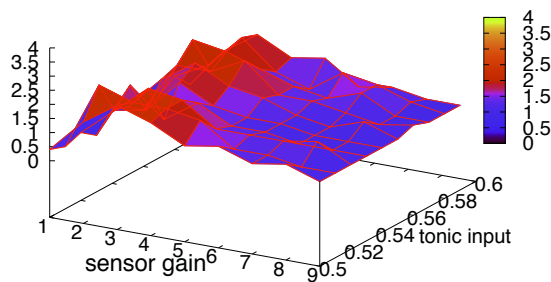


Figure 10: Numbers of behavioral attractors with a scale-free network (hidden nodes = 500)

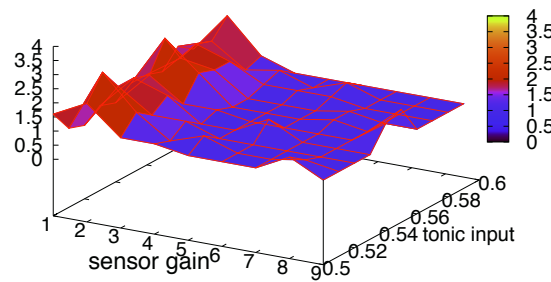


Figure 12: Number of behavioral attractors with a small-world network (hidden nodes = 500)

## Result

Figure 5 and 6 indicate time series of the robot movements. The robot mimics the motions of a natural snake. Throughout the experiments, the behaviors can be classified into a maximum of four movements: forward crawling (forward traveling waves), backward crawling (backward traveling waves), bending (stationary waves), and phase-shifted bending.

The average estimated behavioral attractors in Experiment (a) (no connections among the oscillators; one-to-one correspondence between oscillators and muscles) are shown in Figure 8 for different tonic inputs and sensor gains. The number of attractors is one for almost all parameter values. The results of Experiments (b.1) and (b.2) are shown in Figure 9 and 10, while those of Experiments (c.1) and (c.2) are shown in Figure 11 and 12, respectively. Distinct peaks appear in the landscapes of the estimated number of attractors in the networks of the parameter space. These results indicate that there are appropriate and inappropriate connectivity between networks and the body for versatile behaviors.

The effect of the sensor gain parameter was investigated at fixed tonic input 0.55. The results for networks without hidden nodes (Experiments (a), (b.1)-(e.1)) and containing 500 hidden nodes (Experiments (a), (b.2)-(e.2)) are shown in Figure 13 and Figure 14, respectively. Peak locations in networks generated by our algorithm and the Watts algorithm are identical, but differences exist between the networks themselves. Significant differences are

found among Experiments a, b.1, c.1, and d.1 (ANOVA, F-value = 4.635,  $p = 0.00391$ ), and among Experiments (a), (b.2), (c.2), and (d.2) at gain = 2 (ANOVA, F-value = 11.730,  $p = 5.68 \cdot 10^{-7}$ ). We consider that the scale-free nature of the network, rather than the average shortest distance between nodes, influences the movement behavior.

## Discussion

We examined several emergent behaviors resulting from interactions between different network structures, using nonlinear oscillators coupled to a musculoskeletal body. The behaviors of the models were diversified by the presence of complex network structures with optimal sensor gains. Because the number of periodic behaviors peaks around a certain sensor gain, we suggest that the interactions will be important to induce the most diverse behaviors.

Since bodily constraints induce stable periodic behaviors in Experiment (a), we consider the musculoskeletal body actuated by polyarticular muscles as an attractor of consistent behaviors. According to Jalan et al. (2005), oscillator complex networks dynamically generate diverse clusters. Therefore, for diverse behaviors to emerge, the combined dynamics of complex network and body must be well-matched. This consideration might be relevant to studies of the autistic brain and general movements.

Taga et al. (1999) identified a U shape development in healthy general movements. Using nonlinear prediction analysis, they re-

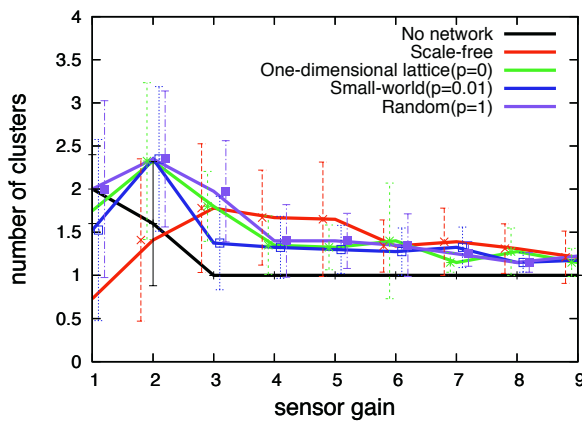


Figure 13: The number of behavioral attractors without a hidden network. Tonic input: 0.55

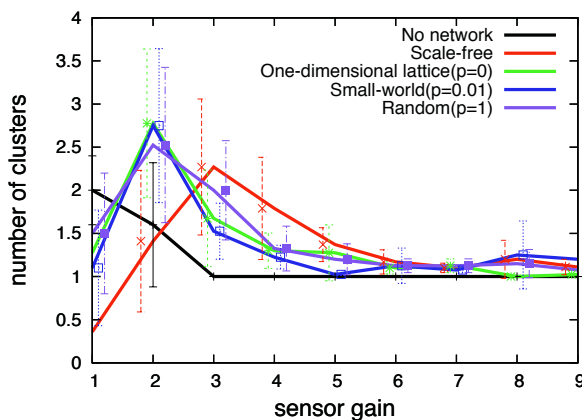


Figure 14: Numbers of behavioral attractors in the existence of each network with a network with 500 hidden nodes. Tonic input: 0.55

ported that movement complexity is high immediately following birth, decreases at around 2 months, and increases thereafter. The simulation results in this paper suggest that behavior becomes simpler when the dynamics are poorly matched, even if the nervous system is correctly characterized by a complex network. Developmental behavioral changes in human infants may manifest from matching of dynamics among the nervous system, the body and the ambient environment.

A recent MRI study found that the white matter in the brains of children with autistic spectrum disorder (ASD) is structurally different from that of their typically developing peers (Wolff et al. (2012); Courchesne et al. (2007)). Specifically, the brain tissue of ASD children displayed fewer long-range connectivities and stronger local connectivities. Motor development in premature infants who are later diagnosed as ASD is also atypical (Karmel et al. (2010)). Hadders-Algra (2008) proposed that the structural differences in the cerebral cortex of individuals with cerebral palsy and ASD cause reduced variability in motor behavior.

Can such structural differences explain the behavioral character-

istics of ASD, such as repetitive behaviors? From a complex systems perspective, repetitive behaviors in ASD may be regarded as a kind of stable state pulled into strong attractors within the dynamics of brain structure, body, and surrounding environment. Namely, an apparent relationship exists between complex cerebral networks and behavioral characteristics. Although we did not provide a direct treatment of this problem in this paper, we believe that our approach will yield theoretical insights into developmental disorders such as ASD.

## Conclusion

We modeled complex networks by nonlinear oscillators connected to a musculoskeletal body model, and conducted simulations in a range of scenarios. Diverse behaviors emerged in the combined network/body system under certain network structures and sensor gains. The results suggest that the network structure of human brains plays an important role in the emergence of diverse behaviors (such as general movements) in early human development. Future work will reveal fine differences between cerebral network structures and the whole body musculoskeletal system of fetuses and infants (Kuniyoshi and Sangawa (2006); Mori and Kuniyoshi (2010)). Such studies will significantly advance our understanding of human behavioral development.

## Acknowledgements

The work reported in this paper has been supported by Grant-in-Aid for Scientific Research on Innovative Areas "Constructive Developmental Science -Revealing the Principles of Development form Fetal Period and Systematic Understanding of Developmental Disorders-" (No. 24119001), Grant-in-Aid for Specially Promoted Research "Constructive Developmental Science Based on Understanding the Process from Neuro-Dynamics to Social Interaction" (No. 24000012) and Grant-in-Aid for Young Scientists (A) "Constructive Developmental Research for Human Development from Fetus to Infant which is Induced by Structural Constraint of the Nervous System, the Body and the Environment" (No. 24680024) from The Ministry of Education, Culture, Sports, Science and Technology, Japan.

## References

- Asai, Y., Nomura, T., and Sato, S. (2000). Emergence of oscillation in a model of weakly coupled two bonhoeffer-van der pol equations. *BioSystems*, 58:239–247.
- Barabasi, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286:509–512.
- Buzsaki, G. and Watson, B. O. (2012). Brain rhythms and neural syntax: implications for efficient coding of cognitive content and neuropsychiatric disease. *Dialogues in clinical neuroscience*, 14(4):345–367.
- Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE transactions on pattern analysis and machine intelligence*, 24(5):603–619.
- Courchesne, E., Pierce, K., Schumann, C. M., Redcay, E., Buckwalter, J. A., Kennedy, D. P., and Morgan, J. (2007). Mapping early brain development in autism. *Neuron*, 56:399–413.
- Hadders-Algra, M. (2008). Reduced variability in motor behaviour: An indicator of impaired cerebral connectivity? *Early Human Development*, 84:787–789.



- Jalan, S., Amritkar, R. E., , and Hu, C.-K. (2005). Synchronized clusters in coupled map networks. i. numerical studies. *Physical Review*, 72.
- Kaneko, K. and Tsuda, I. (2003). Chaotic itinerancy. *Chaos*, 13:926–936.
- Karmel, B. Z., Gardner, J. M., Meade, L. S., Cohen, I. L., London, E., Flory, M. J., Lennon, E. M., Miroshnichenko, I., Rabinowitz, S., Parab, S., Barone, A., and Harin, A. (2010). Early medical and behavioral characteristics of nicu infants later classified with asd. *Pediatrics*, 126:1–11.
- Kuniyoshi, Y. and Sangawa, S. (2006). Early motor development from partially ordered neural-body dynamics: experiments with a cortico-spinal-musculo-skeletal model. *Biological Cybernetics*, 95(6):589–605.
- Kuniyoshi, Y. and Suzuki, S. (2004). Dynamic emergence and adaptation of behavior through embodiment as coupled chaotic field. *Proceedings of 2004 IEE/RSJ International conference on intelligent robots and systems*, pages 2042–2048.
- Mori, H. and Kuniyoshi, Y. (2010). A human fetus development simulation: Self-organization of behaviors through tactile sensation. In *IEEE 9th International Conference on Development and Learning (ICDL 2010)*, pages 82–97.
- Niiyama, R., Nagakubo, A., and Kuniyoshi, Y. (2007). Mowgli: A bipedal jumping and landing robot. In *2007 IEEE International Conference on Robotics and Automation*, pages 2546–2551.
- Prechtl, H. F. R. (2001). General movement assessment as a method of developmental neurology: new paradigms and their consequences. *Developmental Medicine & Child Neurology*, 43:836–842.
- Smith, R. (2001). Open dynamics engine. <http://www.ode.org/>.
- Spittle, A. J., Brown, N. C., Doyle, L. W., Boyd, R. N., Hunt, R. W., Bear, M., and Inder, T. E. (2008). Quality of general movements is related to white matter pathology in very preterm infants. *Pediatrics*, 121:e1184.
- Sporns, O. (2012). *Discovering the human connectome*. The MIT Press.
- Taga, G., Tanaka, R., and Konishi, Y. (1999). Analysis of general movements of infants towards understanding of developmental principle for motor control. *Proceeding of IEEE International Conference on Systems, Man, and Cybernetics*, pages 678–683.
- Tsuda, I., Fujii, H., Tadokoro, S., Yasuoka, T., and Yamaguti, Y. (2004). Chaotic itinerancy as a mechanism of irregular changes between synchronization and desynchronization in a neural network. *Journal of integrative neuroscience*, 3(2):159–182.
- van den Hauvel, M. P. and Sporns, O. (2011). Rich-club organization of the human connectome. *The Journal of Neuroscience*, 31:15775–15786.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393:440–442.
- Wolff, J. J., Gu, H., Gerig, G., Ellison, J. T., Styner, M., Gouttard, S., Botteron, K. N., Dager, S. R., Dawson, G., Estes, A. M., Evans, A. C., Hazlett, H. C., and Kostopoulos, P. (2012). Differences in white matter fiber tract development present from 6 to 24 months in infants with autism. *The american journal of psychiatry*, 169:589–600.

# Collective Dynamics and Homeostatic Emergence in Complex Adaptive Ecosystem

Dharani Punithan and RI (Bob) McKay

Structural Complexity Laboratory,  
College of Engineering, School of Computer Science and Engineering,  
Seoul National University, South Korea  
{punithan.dharani,rimsnucse}@gmail.com

## Abstract

We investigate the behaviour of the daisyworld model on an adaptive network, comparing it to previous studies on a fixed topology grid, and a fixed small-world (Newman-Watts (NW)) network. The adaptive networks eventually generate topologies with small-world effect behaving similarly to the NW model – and radically different from the grid world. Under the same parameter settings, static but complex patterns emerge in the grid world. In the NW model, we see the emergence of completely coherent periodic dominance. In the adaptive-topology world, the systems may transit through varied behaviours, but can self-organise to a small-world network structure with similar cyclic behaviour to the NW model.

## Introduction

In this paper, we examine connectivity changes in a complex adaptive ecosystem based on the daisyworld model, combining coupled map lattice (CML) and complex adaptive network models. Daisyworld, proposed by Watson and Lovelock (1983), is a simple mathematical system demonstrating planetary homeostasis – self-regulation of the environment by biota and self-sustainability of life through interaction with the environment. Daisyworld topologies in the literature are static, with only local connections (Wood et al., 2008). In our previous work (Punithan et al., 2011; Punithan and McKay, 2013), we have investigated ecological homeostasis in preconstructed static topologies with local and non-local long range couplings – small-world networks. But complex networks in nature and society are adaptive, in that they exhibit feedback between the local dynamics of nodes (state) and the evolution of the topological structure (Gross and Blasius, 2008; Gross and Sayama, 2009). Examples include genetic, neural, immunity, ecological, economic and social networks, complex game interactions etc.

The topology of our ecosystem evolves in response to local habitat states, and the evolved topology in turn impacts the habitat states. Our adaptive and self-maintaining ecosystem, based on CML consists of a set of diffusively coupled habitats incorporating logistic growth of life with bi-directional biota-environment influences. Thus our ecosystem incorporates three kinds of feedback:

1. Life-environment feedback via the daisyworld model
2. State-topology feedback via an adaptive network model
3. Density-growth feedback via a logistic growth model

The topology of the our ecosystem evolves with a simple local rule – a frozen habitat is reciprocally linked to an active habitat – and self-organises to complex topologies with small-world effect. In this paper, we focus on the emergent collective phenomena and properties that arise in egalitarian small-world ecosystems, constructed from a large number of interacting adaptively linked habitats.

## Background

Our model has three feedback loops determining its dynamics. We next detail the relevant background.

**Daisyworld (homeostatic self-regulation of the environment by the biota)** Daisyworld (Watson and Lovelock, 1983) is an imaginary planet where only two types of species live – black and white daisies. These biotic components interact stigmergically via an abiotic component – temperature. The different colours of the daisies influence the albedo (reflectivity) of the planet. In the beginning, the atmosphere of the daisyworld is cooler and only black daisies thrive as they absorb all the energy. As the black daisy population expands, it warms the planet. When it is too warm for black daisies to survive, white daisies start to bloom since they reflect all the energy back into space. As the white daisy cover spreads, it cools the planet. When it is too cold for the survival of white daisies, again black daisies thrive. This endless cycle, owing to the bi-directional feedback loop between life and the environment, self-regulates the temperature and thereby allows life to persist.

**Adaptive Networks (dynamics on the network interacting with dynamics of the network)** In most real-world networks, the topology itself is a dynamical system which changes in time and in response to the dynamics of the states of the nodes (dynamics of the network). The evolved topology in turn influences the dynamics of the states of the nodes

(dynamics *on* the network), creating a feedback loop between the dynamics of the nodes and the evolution of the topology. Networks exhibiting such a feedback loop (mutual evolution of structure and state values) are called adaptive or coevolutionary networks (Gross and Blasius, 2008; Gross and Sayama, 2009). In road networks, the topology of the road influences the traffic flow, while traffic congestion influences the construction of new roads. In the vascular system, the topology of the blood vessels controls blood flow, while restrictions in blood flow influence the formation of new arteries (arteriogenesis). Numerous other examples are discussed in Gross and Blasius (2008).

### Logistic Growth Model (density-dependent growth rate)

The discretised logistic growth model (Verhulst model) is key to population ecology.

$$P_{(t+1)} = P_t \left[ 1 + r \left( 1 - \frac{P_t}{\kappa} \right) \right]; \quad r > 0, \kappa > 0 \quad (1)$$

where  $P \in [0, \kappa]$  is the population size (at times  $t$  and  $t+1$ ),  $r$  is the intrinsic growth rate (bifurcation parameter),  $\kappa$  is the carrying capacity (maximum sustainable population beyond which  $P$  cannot increase). The parameter  $r$  amplifies population growth and the component  $[1 - \frac{P_t}{\kappa}]$  dampens the growth due to over crowding. Thus population density self-regulates population growth rate. It is also well-known that chaos emerges from this growth model (May, 1976) in spite of the built-in regulatory mechanism.

**Coupled Map Lattice** The coupled map lattice (Kaneko, 1985, 1992; Kaneko and Tsuda, 2001) incorporates discrete time evolution (map) in a discrete space (lattice or network) as in cellular automata (CA), but takes continuous state values as in partial differential equation (PDE) models. CML is governed by the temporal nonlinear reaction (maps -  $f$ ) and the spatial diffusion (coupling -  $\epsilon$ ).

If  $f(x)$  is a reaction function of a dynamical variable ( $x$ ), the update of the variable is computed by combining that reaction with discrete Laplacian diffusion. For a regular network with Moore neighbourhoods ( $k = 8$ ), the update of  $x$  is computed as:

$$\begin{aligned} x_{(i,j,t+1)} = f & \left( (1 - \epsilon)x_{(i,j,t)} + \frac{\epsilon}{8} \left[ x_{(i+1,j,t)} \right. \right. \\ & + x_{(i-1,j,t)} + x_{(i,j+1,t)} + x_{(i,j-1,t)} \\ & + x_{(i-1,j-1,t)} + x_{(i+1,j-1,t)} \\ & \left. \left. + x_{(i-1,j+1,t)} + x_{(i+1,j+1,t)} \right] \right) \end{aligned} \quad (2)$$

where  $x_{(i,j,t)}$  is the spatio-temporal distribution of a dynamical variable,  $\epsilon \in [0, 1]$  is the coupling parameter (diffusion rate),  $k$  is the number of interacting neighbours,  $f(x'_{(i,j,t)})$  is a local non-linear function and  $x'_{(i,j,t)}$  is the value after diffusion.

Denoting the set of neighbours of  $(i, j)$  as  $\langle l, m \rangle$ , we can simplify equation 2 to:

$$x_{(i,j,t+1)} = f \left( (1 - \epsilon)x_{(i,j,t)} + \frac{\epsilon}{k} \sum_{\langle l,m \rangle}^k x_{(l,m,t)} \right) \quad (3)$$

**Small-world Phenomena** The co-occurrence of high clustering (as in regular networks) and low characteristic path length (as in random networks) define a small-world structure (Watts and Strogatz, 1998). These small-world network properties, giving rise to the well-known “six degrees of separation” phenomenon (Milgram, 1967), are quantified by two statistical measures: the clustering co-efficient (measuring local cliquiness)  $\mathcal{C}$ , and the characteristic path length  $\mathcal{L}$  (measuring global connectedness). Their average values  $\bar{\mathcal{C}}$  and  $\bar{\mathcal{L}}$  for a network with  $n$  nodes are defined by:

$$\bar{\mathcal{C}} = \frac{1}{n} \sum_{v=1}^n \frac{|E(\Gamma_v)|}{\binom{k_v}{2}} \quad (4)$$

where  $\Gamma_v$  is the neighbourhood of a node  $v$ ,  $|E(\Gamma_v)|$  is the number of actual links in the neighbourhood of  $v$ ,  $k_v$  is the number of nodes in the subnetwork  $\Gamma_v$  and  $\binom{k_v}{2}$  is the number of possible links in  $\Gamma_v$ ; and

$$\bar{\mathcal{L}} = \frac{1}{\binom{n}{2}} \sum_{u=1}^n \sum_{v>u}^n d_{uv} \quad (5)$$

where  $d_{uv}$  is the shortest path between a pair of nodes  $u, v$ .

**Degree Distribution** The degree of a node is the number of neighbours it is connected to. The degree distribution is defined as the normalised frequency distribution of degrees over the whole network. The degree distribution of a network is a simple property which helps to classify networks. The regular network with Moore neighbourhoods have the same degree ( $k = 8$ ) for all the nodes. The degree distribution of small-world networks ( $p$  in the small-world regime (Punithan and McKay, 2013) follows a Poisson distribution with exponential tail. Networks in which most nodes have approximately the same number of neighbours are known as “egalitarian” networks (Buchanan, 2003).

## Model

Our ecosystem is a complex dynamic system in which the continuous state habitats diffusively interact with their neighbours (coupled), evolve in discrete time (map) and are distributed on a discrete space (lattice). Initially, we construct a 2-lattice with Moore neighbourhoods and periodic boundary conditions. Each point in the lattice represents a habitat with a maximum carrying capacity of 10,000 daisies. Each habitat in our ecosystem is a system. The elements

such as life (black and white daisies) and environment (temperature) are interconnected and interdependent via reinforcing and balancing feedback loops. At each succession of a habitat, we compute the population of black and white daisies, and the temperature, based on equation 3.

Table 1: Daisyworld Parameter Settings

Parameter	Value
Number of habitats ( $N \times N$ )	$100 \times 100$
Heat capacity ( $C$ ) $Wm^{-2}K^{-1}$	2500
Diffusion constant ( $D_T$ ) $Wm^{-2}K^{-1}$	500
Stefan-Boltzmann constant ( $\sigma_B$ ) $E^{-8}Wm^{-2}K^{-4}$	5.67
Luminosity ( $L$ )	1
Solar Insolation ( $S$ ) $Wm^{-2}$	864.65
Noise Level ( $K$ )	0.001
Opt. temp of black daisies $T_{opt_b}$ (K)	284.5
Opt. temp of white daisies $T_{opt_w}$ (K)	306.5
Carrying capacity ( $\kappa$ )	10000
Dispersion rate of daisies ( $D_c$ )	0.2
Natural rate of increase ( $r$ )	1

### Albedo:

The albedo ( $A$ ) at a lattice point  $(i, j)$  and at time  $(t)$  is

$$A_{(i,j,t)} = A_b(\alpha_b)_{(i,j,t)} + A_w(\alpha_w)_{(i,j,t)} + A_g(\alpha_g)_{(i,j,t)} \quad (6)$$

i.e. the average of the albedos  $A_b$  of ground covered by black daisies,  $A_w$  of ground covered by white daisies and  $A_g$  of bare ground, weighted by  $\alpha_b, \alpha_w, \alpha_g (= 1 - \alpha_w - \alpha_b) \in [0, 1]$ , the relative areas occupied by black, white daisies and bare ground at time  $t$ . We assume that  $A_w > A_g > A_b$ , with corresponding values of 0.75, 0.5, 0.25.

### Growth:

The growth curve of daisies ( $\beta_c$ ) is an inverted parabola:

$$\beta_c(T_{(i,j,t)}) = \max \left( 1 - \left[ \frac{(T_{opt_c} - T_{(i,j,t)})^2}{17.5^2} \right], 0 \right) \quad (7)$$

$T_{(i,j,t)}$  is the local temperature and  $T_{opt_c}$  is the optimal temperature of the species. The optimal temperature of the daisies depends on their petal colour 'c' (phenotype). The optimal temperature for black daisies is lower than for white; the mean optimal temperature is assumed to be 295.5K.

### Temperature:

The temperature ( $T_{(i,j,t+1)}$ ) is computed as the sum of temperature after Laplacian diffusion ( $T'_{(i,j,t)}$ ), the difference between solar absorption and heat radiation incorporating  $T'_{(i,j,t)}$ , and Gaussian white noise:

$$T_{(i,j,t+1)} = g \left( (1 - D)T_{(i,j,t)} + \frac{D}{k} \sum_{<l,m>} T_{(l,m,t)} \right) \quad (8)$$

where  $T_{(i,j,t)}$  is local temperature,  $<l, m>$  represents the set of neighbours of  $(i, j)$  and  $D = D_T/C$  is the thermal diffusion constant normalised by heat capacity  $C$ .  $g(T'_{(i,j,t)})$  is the temperature update function (Wood et al., 2008), in which  $T'_{(i,j,t)}$  is the temperature after diffusion:

$$g(T'_{(i,j,t)}) = T'_{(i,j,t)} + \xi + \left[ \frac{(SL(1 - A_{(i,j,t)}) - \sigma_B(T'_{(i,j,t)})^4)}{C} \right] \quad (9)$$

where  $S$  is the solar constant,  $L$  is the luminosity,  $A_{(i,j,t)}$  is the albedo,  $\sigma_B$  is the Stefan-Boltzmann constant and  $\xi$  is additive Gaussian white noise (with mean 0 and standard deviation 1.0) multiplied by the noise level (NL).

### Population size:

The local population update depends on dispersion, density-dependant growth rate and the feedback coefficient:

$$P_{c(i,j,t+1)} = h \left( (1 - D_c)P_{c(i,j,t)} + \frac{D_c}{k} \sum_{<l,m>} P_{c(l,m,t)} \right) \quad (10)$$

where  $P_{c(i,j,t)}$  is the population size at location  $(i, j)$  and time step  $t$ ,  $D_c$  is the fraction of the population being dispersed to its neighbours,  $c$  stands for colour of daisies and  $k$  is number of neighbours.  $h(P'_{c(i,j,t)})$  is the population growth function and  $P'_{c(i,j,t)}$  is population size after dispersion:

$$h(P'_{c(i,j,t)}) = P'_{c(i,j,t)} \left[ 1 + r \left( \beta_c(T_{(i,j,t)}) - \frac{P'_{c(i,j,t)}}{\kappa} \right) \right] \quad (11)$$

where  $r$  is population increase rate,  $\beta_c(T_{(i,j,t)})$  is the feedback due to temperature and  $\kappa$  is the carrying capacity.

### The Small-world Network Model

Small-world networks can be modelled in various ways – Watts and Strogatz (1998) (WS) model, Newman and Watts (1999) (NW) model, etc. Although the WS model was a breakthrough in network science, it may not guarantee connectivity owing to the rewiring process – deleting connections in the underlying network may result in disjoint nodes. Hence we use the later NW model, where we only add long-range connections. For each connection in the underlying ecosystem, a new reciprocal connection is added to a randomly chosen non-local habitat with probability  $p \in [0, 1]$ . In this work, we have chosen  $p = 0.05$ , since it is in the small-world regime and has proven to have interesting dynamics (Punithan and McKay, 2013).



## Adaptive Network Model

Adaptive networks are a class of dynamical networks whose topologies and states coevolve. Dynamic Linking (DL) is the key feature of adaptive networks, and can be modelled in a number of ways:

1. Active nodes grow and inactive nodes lose links
2. Active nodes lose links and inactive nodes grow them
3. Nodes never lose links; the network evolves by either:
  - (a) Adding new links to active nodes from inactive nodes
  - (b) Adding new links to inactive nodes from active nodes
  - (c) Adding reciprocal links between active and inactive nodes

By means of DL, we model the topology of our ecosystem itself as a dynamical system, changing in time according to a simple local rule (dynamics of networks). Each habitat, representing a dynamical system (dynamics on networks), is dynamically coupled according to the evolved topology. In our ecosystem, we never remove connections between habitats; we add new reciprocal connections between frozen habitats and active habitats (i.e. method 3c). This simple rule gives rise to a complex topology.

In our model, only black and white daisies disperse via both local and long-range connections, created either statically or dynamically (by water, air, animal pollinator transport etc.), while temperature diffuses only locally.

## Experiments

### Experiment Settings

The habitats are randomly initialised with a population size in  $[0, 100]$  for both species and with the temperature in  $[280, 310]K$ . We permit both species of daisies to coexist, hence we allow an overlap of 10% in the growth response to temperature. The overlap chosen determines the optimal temperature values of daisies. The parameter and their values are described in Table 1.

We have investigated daisyworld phenomenon in three different topological scenarios:

1. We start with an ecosystem where habitats are only locally connected (regular CML with Moore neighbourhood).
2. We add random non-local links to the underlying regular lattice, which introduces small-world effects in ecosystem (Newman-Watts model in CML)
3. Each frozen node in the underlying regular lattice is dynamically and reciprocally linked to a randomly chosen active node (adaptive CML).

A node is said to be frozen when its local dynamics are static – black and white daisies maintain the same population size

for six consecutive epochs. The links added either statically (NW) or dynamically (adaptive) are reciprocal links (mutual links). We ran 25 realisations of each network model, and present a typical example of a run of each model. Scenarios 1 and 2 were previously analysed in Punithan et al. (2011) and Punithan and McKay (2013), though with different overlaps (0% and 5%) respectively.

## Visualisations

We capture snapshots from the evolution of daisyworld to inspect its spatio-temporal dynamics. Each snapshot represents the population structure of the ecosystem at the particular epoch. As it is impractical to show all the snapshots over 5,000 epochs, we plot the temporal dynamics of daisy populations and temperature at a particular habitat as well as the temporal dynamics of the average daisy populations and temperature of the whole ecosystem. These plots reflect the behaviour of the daisyworld.

In the visualisations, a habitat is shown as black if black daisies alone occupy that habitat and correspondingly for white. If both daisies coexist at a habitat but black dominates, it is shown as dark grey; if white dominates, it is presented as light grey; and if the populations are equal, it is represented as medium grey.

## Results

### I. Daisyworld with Static Local Couplings (Regular Networks)

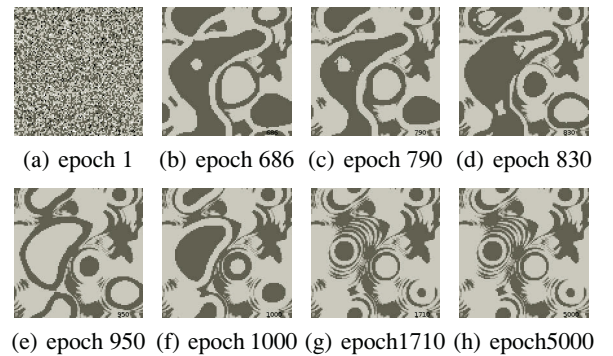
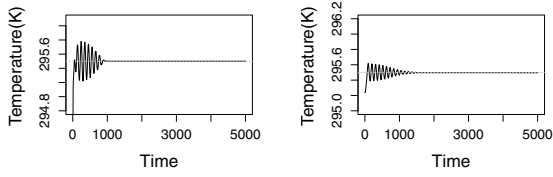


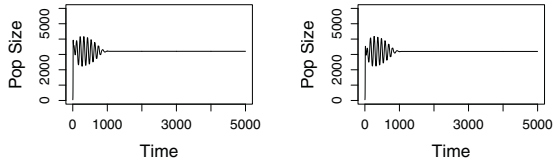
Figure 1: Regular CML:  $D = 0.2$  and  $NL = 0.001$  in 2D  $100 \times 100$

With only local couplings, we observe the formation of complex static patterns. The whole ecosystem freezes after epoch 1710. This scenario is clearly seen in the snapshots (Figure 1), in global population dynamics (Figure 4) and in global temperature dynamics (sub Figure 2 (b)). The local population dynamics (Figure 3) and local temperature (sub Figure 2 (a)) at a typical habitat (57, 50) shows that the dynamics freezes even quicker (epoch 1055). All trajectories show initial fluctuations but evolve to complete stationarity.



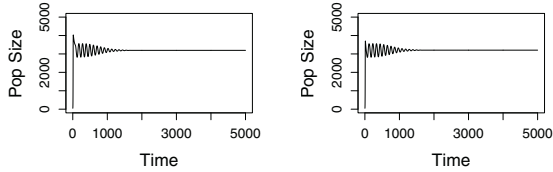
(a) Local Temp. at habitat (57, 50) (b) Global Surface Temp.

Figure 2: Regular CML: Temperature dynamics  $D = 0.2$  and  $NL = 0.001$



(a) Black Daisy Population (b) White Daisy Population

Figure 3: Regular CML: Local population dynamics at habitat (57, 50);  $D = 0.2$  and  $NL = 0.001$



(a) Black Daisy Abundance (b) White Daisy Abundance

Figure 4: Regular CML: Global population dynamics;  $D = 0.2$  and  $NL = 0.001$

## II. Daisyworld with Static Local and Non-local Couplings (Small-World Network)

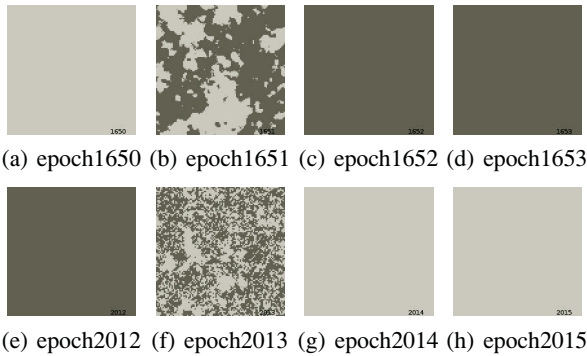
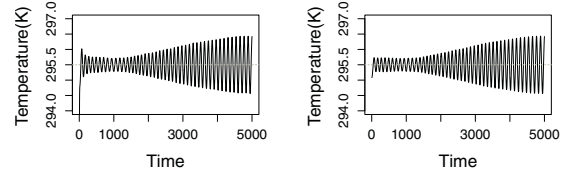
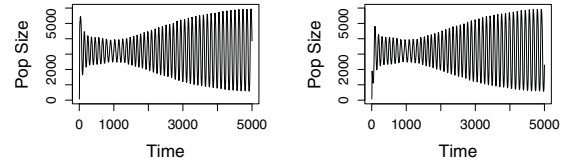


Figure 5: Newman-Watts ( $p = 0.05$ ) :  $D = 0.2$  and  $NL = 0.001$  in 2D  $100 \times 100$  Small-world CML



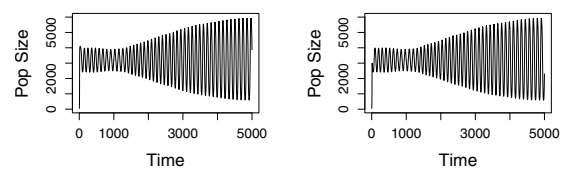
(a) Local Temp. at habitat (52, 58) (b) Global Surface Temp.

Figure 6: Newman-Watts CML ( $p = 0.05$ ): Temperature dynamics  $D = 0.2$  and  $NL = 0.001$



(a) Black Daisy Population (b) White Daisy Population

Figure 7: Newman-Watts CML ( $p = 0.05$ ): Local population dynamics at habitat (52, 58);  $D = 0.2$  and  $NL = 0.001$



(a) Black Daisy Abundance (b) White Daisy Abundance

Figure 8: Newman-Watts CML ( $p = 0.05$ ): Global population dynamics;  $D = 0.2$  and  $NL = 0.001$

Each dynamic unit (habitat) is coupled through a small-world topology initialised through the NW mechanism. The topology remains static but the states of the habitats change dynamically. This ecosystem exhibits a periodic behaviour (Figure 5). The cyclic behaviour is understood by observing the trajectories in temperature dynamics (Figure 6), local population dynamics (Figure 7) and global population dynamics (Figure 8). This shows that a small change in the underlying topology drastically influences the dynamical properties of the ecosystem, and the transition is very abrupt. Within a single time frame, black dominance may change to white dominance, or vice versa (Figure 5).

## III. Daisyworld with Dynamic Local and Non-local Couplings (Adaptive Network)

The local dynamical linking rule (when a node becomes frozen, we allow that habitat to reciprocally connect to a random active habitat) generates topologies with small-world

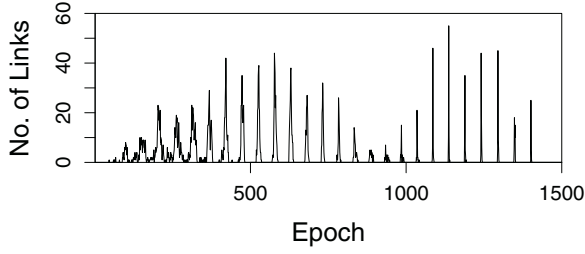
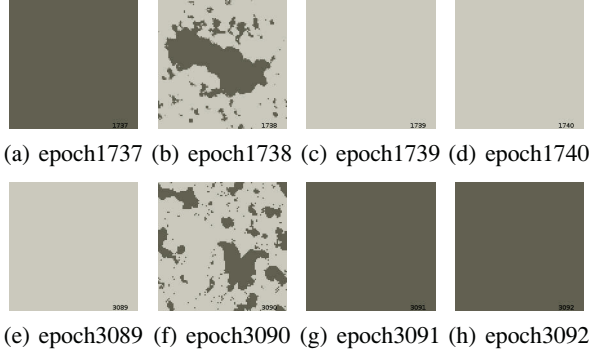


Figure 9: Adapted Reciprocal Links


 Figure 10: Adaptive CML :  $D = 0.2$  and  $NL = 0.001$  in 2D  $100 \times 100$ 

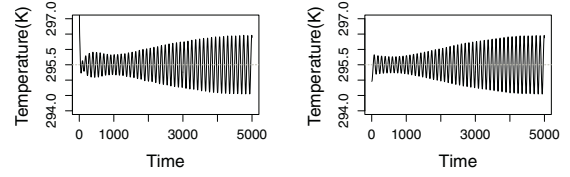
effect (low characteristic path length (CP)) – here we show topologies with high clustering coefficient (CC) (approximately 50% of runs) to compare to NW-CML ( $p = 0.05$ ). It can also generate topologies with low CC. It depends on the random initialization of the temperature of each habitat and random dynamic linking. Typically we observe periodic behaviour (Figure 10) similar to NW-CML. In the corresponding time series plots, the dynamics of both local and global temperature (Figure 11), local population (Figure 12) and global population (Figure 13) exhibit cyclic behaviour. The dynamically adapted reciprocal links are shown in Figure 9.

### Why are NW-CML and adaptive CML similar?

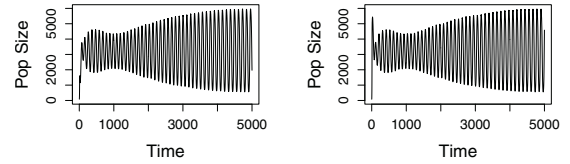
Table 2: Typical Clustering Coefficients and Characteristic Path Lengths

Model	$\bar{C}/C_{\text{regular}}$	$\bar{L}/L_{\text{regular}}$
NW-CML	0.837	0.185
Adaptive CML	0.837	0.189

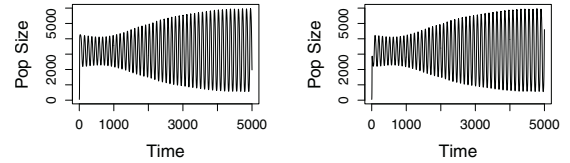
We saw very similar limit behaviours from NW-CML (Subsection II) and adaptive CML (Subsection III). We can gain understanding through analysing the topological quantifiers (degree distribution, clustering coefficient and characteristic path length) for their network topologies.



(a) Local Temp. at habitat (b) Global Surface Temp. (56, 57)

 Figure 11: Adaptive CML: Temperature dynamics  $D = 0.2$  and  $NL = 0.001$ 


(a) Black Daisy Population (b) White Daisy Population

 Figure 12: Adaptive CML: Local population dynamics at habitat (56, 57);  $D = 0.2$  and  $NL = 0.001$ 


(a) Black Daisy Abundance (b) White Daisy Abundance

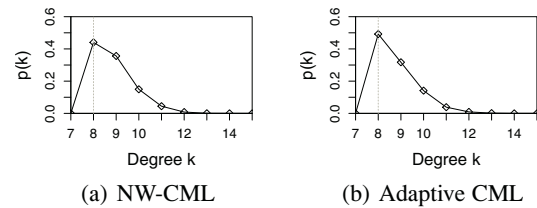
 Figure 13: Adaptive CML: Global population dynamics;  $D = 0.2$  and  $NL = 0.001$ 


Figure 14: Degree Distribution

The degree distribution of the small-world CML constructed via the NW model in sub section II ranges over  $[8, 14]$  and has an exponential tail reaching zero, as shown in the Figure 14(a). The degree distribution reached by adaptive CML in sub section III, which dynamically linked with reciprocal links, also ranges over  $[8, 14]$  and has an expo-

nential tail reaching zero as shown in the Figure 14(b).

The CC for NW-CML, and that for the final epoch of adaptive CML are almost the same, as are CPs. Figure 14 and Table 2 show the results. The finally-converged adaptive CML is an egalitarian small-world network. This is why we observe a drastic change in the dynamics of the system compared to the regular lattice. It also shows that the topology, constructed statically or dynamically, influences the collective behaviour of the system: relatively small changes in the linkage structure can generate vastly different dynamics.

Table 3: Average( $\pm$ Std.Dev.) Clustering Coefficients and Characteristic Path Lengths

Model	$\bar{C}/C_{\text{regular}}^-$	$\bar{L}/L_{\text{regular}}^-$
NW-CML	$0.8412 \pm 0.0024$	$0.1868 \pm 0.001$
Adaptive CML	$0.8928 \pm 0.0639$	$0.2204 \pm 0.0392$

The sections II and III illustrate typical scenarios of NW-CML and adaptive CML. Table 3 shows averages over 25 realisations of adaptive CML model and 25 of NW-CML ( $p = 0.05$ ) model. We ran 100 realizations of adaptive CML and picked 25 that fell in small-world regime (Punithan and McKay, 2013) for comparison purposes – CC in  $[0.98, 0.7]$  and CP in  $[0.3, 0.16]$ . CC and CP are normalised by the values for a regular lattice as proposed in Watts and Strogatz (1998).

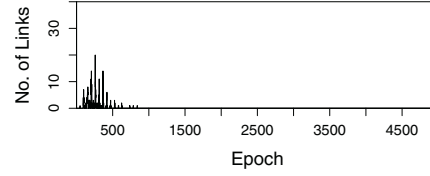
## Topological Evolution

Table 4: Clustering Coefficients and Characteristic Path Lengths

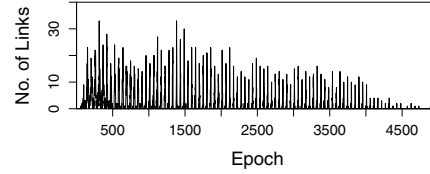
Adaptation	$\bar{C}/C_{\text{regular}}^-$	$\bar{L}/L_{\text{regular}}^-$
Quick	0.977	0.295
Slow	0.837	0.238

The evolution of the topology continues until the stationary attractor (frozen local dynamics) of all habitats reach a dynamical attractor – here a limit cycle. Some samples adapt quickly, reaching a stable topology around the 500<sup>th</sup> epoch (Figure 15 (a) – Quick Adaptation), while a few evolve almost until the 5000<sup>th</sup> epoch (Figure 15 (b) – Slow Adaptation). Their degree distribution (Figure 16), clustering coefficient and characteristic path (Table 4) show that both evolve to small-world networks, although at different rates.

The collective dynamics in both quick and slow adaptations (Figures 18 and 19) show the shift in dominance is not so abrupt as in Figures 5 and 10 (see epoch nos.). The emergent property – the temperature cycles – depicted in Figure 17 (in both adaptations) have different limit ranges. In sub Figure 17 (b), the shift in dominance takes more time initially, eventually speeding up owing to the increasing limit height. The abruptness is clear in the corresponding global temperature plots – compare the limits in sub fig-

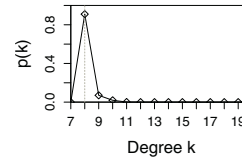


(a) Quick Adaptation

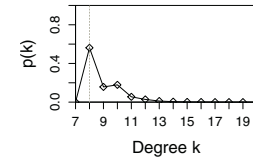


(b) Slow Adaptation

Figure 15: Adapted Reciprocal Links

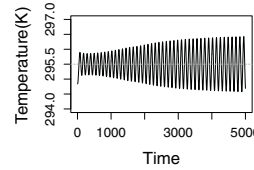


(a) Quick Adaptation

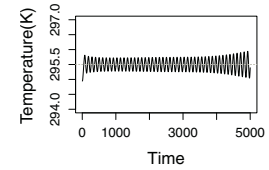


(b) Slow Adaptation

Figure 16: Degree Distribution



(a) Quick Adaptation



(b) Slow Adaptation

Figure 17: Global Temperature dynamics  $D = 0.2$  and  $NL = 0.001$

ures 6 (b), 11 (b), 17 (a) and 17 (b). The degree distribution shows that relatively few reciprocal links are added (sub figure 16 (a)). This confirms even a few long-distance links – if they are the right links – lead to drastic behaviours changes.

## Conclusion

We have analysed the connectivity changes in a complex adaptive ecosystem combining life-environment, state-topology and density-growth feedback loops. The results illustrate the capacity of the adaptive ecosystem to self-organise to a complex ecosystem (small-world network)



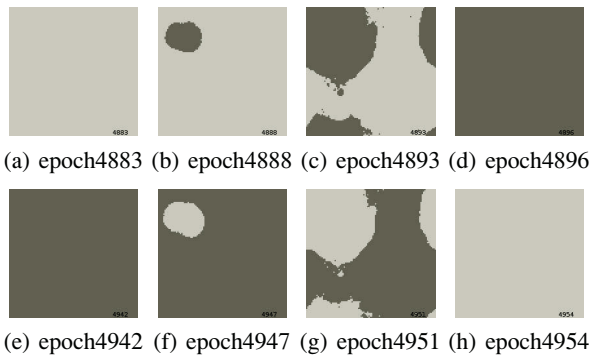


Figure 18: Quick Adaptation :  $D = 0.2$  and  $NL = 0.001$  in 2D  $100 \times 100$

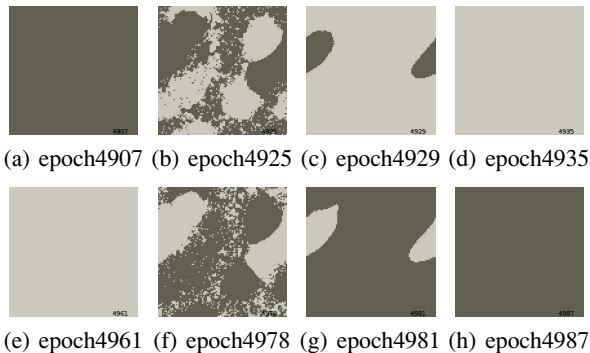


Figure 19: Slow Adaptation :  $D = 0.2$  and  $NL = 0.001$  in 2D  $100 \times 100$

through a simple dynamical rule – frozen habitats (nodes) gain reciprocal non-local neighbours (links). This ecosystem exhibits similar behaviour to contagion systems such as memes or virus – cyclic behaviours – without any external intervention, requiring only the adding of new reciprocal connections under certain locally-defined conditions.

Even a small change in the connectivity, with almost no effect on the mean degree of the ecosystem, leads to a drastic behaviour change from the grid network. It is much more like the real-world behaviour we see in social systems (seasonal rise and fall of fads), economic systems (booms and busts) etc.. This “small cause, large effect” behaviour draws analogies with popular metaphors black swan (low probable but high-impact events) (Taleb, 2010), butterfly effect (sensitive dependence on initial conditions) (Hilborn, 2004) and tipping point (little things make a big difference) (Gladwell, 2006). Though the collective dynamics change in varying ways, we still observe the emergent property – self-regulation of the temperature at around  $295.5K$ .

## Acknowledgements

This research was supported by the Basic Science Research Program of the National Research Foundation of Korea

(NRF) funded by the Ministry of Education, Science and Technology (Project No. NRF-2012R1A1A2004841). The Institute for Computer Technology (ICT) at Seoul National University provided research facilities for the study. We thank Ilun Science and Technology Foundation for their generous support of Dharani Punithan.

## References

- Buchanan, M. (2003). *Nexus: small worlds and the groundbreaking theory of networks*. WW Norton.
- Gladwell, M. (2006). *The tipping point: How little things can make a big difference*. Little, Brown.
- Gross, T. and Blasius, B. (2008). Adaptive coevolutionary networks: a review. *Journal of the Royal Society Interface*, 5(20):259–271.
- Gross, T. and Sayama, H. (2009). Adaptive networks. *Adaptive Networks*, pages 1–8.
- Hilborn, R. C. (2004). Sea gulls, butterflies, and grasshoppers: A brief history of the butterfly effect in nonlinear dynamics. *American Journal of Physics*, 72:425.
- Kaneko, K. (1985). Spatiotemporal intermittency in coupled map lattices. *Progress of Theoretical Physics*, 74(5):1033–1044.
- Kaneko, K. (1992). Overview of coupled map lattices. *Chaos*, 2(3):279–282.
- Kaneko, K. and Tsuda, I. (2001). *Complex Systems: Chaos and Beyond: a Constructive Approach with Applications in Life Sciences*. Springer Verlag.
- May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261(5560):459–467.
- Milgram, S. (1967). The small world problem. *Psychology today*, 2(1):60–67.
- Newman, M. E. J. and Watts, D. J. (1999). Renormalization group analysis of the small-world network model. *Physics Letters A*, 263(4-6):341–346.
- Punithan, D., Kim, D. K., and McKay, R. I. (2011). Daisyworld in two dimensional small-world networks. *Database Theory and Application, Bio-Science and Bio-Technology*, pages 167–178.
- Punithan, D. and McKay, R. I. (2013). Phase transitions in two-dimensional daisyworld with small-world effects: A study of local and long-range couplings. *Future Generation Computer Systems*.
- Taleb, N. N. (2010). *The black swan: The impact of the highly improbable*. Random House Trade Paperbacks.
- Watson, A. J. and Lovelock, J. E. (1983). Biological homeostasis of the global environment: The parable of daisyworld. *Tellus B*, 35(4):284–289.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature*, 393(6684):440–442.
- Wood, A. J., Ackland, G. J., Dyke, J. G., Williams, H. T. P., and Lenton, T. M. (2008). Daisyworld: a review. *Reviews of Geophysics*, 46(1).

# Cellular Automata Coevolution of Update Functions and Topologies: A Tradeoff between Accuracy and Speed

Christian Darabos<sup>1†</sup>, Craig O. Mackenzie<sup>1</sup>, Marco Tomassini<sup>2</sup>, Mario Giacobini<sup>3</sup>, and Jason H. Moore<sup>1</sup>

<sup>1</sup>The Geisel School of Medicine at Dartmouth College, Hanover, NH 03755, USA

<sup>2</sup>Information Systems Department, Faculty of Business and Economics, University of Lausanne, Switzerland

<sup>3</sup>Computational Epidemiology Group at Department of Veterinary Sciences and  
Complex Systems Unit at Molecular Biotechnology Center, University of Torino, Italy

<sup>†</sup>Christian.Darabos@dartmouth.edu

## Abstract

Biological organisms have the ability to develop novel phenotypes in response to environmental changes. When several traits are evolved simultaneously or as a result of one another, we talk of coevolution. Cellular Automata (CAs) have been successfully used to artificially evolve problem specific update functions. The resulting CAs are, however, much slower and more sensitive to perturbations than those with an evolved underlying topology and fixed uniform update rule. Unfortunately, these are not nearly as accurate, and suffer from scaling up the total number of cells. We propose a hybrid paradigm that simultaneously coevolves the supporting network and the update functions of CAs. The resulting systems combine the higher fitness and performance of the update evolution and the robustness properties and speed of the topology evolution CAs. Moreover, these systems seem to perform better as the size of the CA scales up, where as single-feature evolution systems are negatively impacted. Coevolution in CAs is an interesting tradeoff between the two single trait evolutions.

## Introduction

In biology, coevolution refers to the concurrent or sequential mutation in organisms driven by changes in a related biological object (Yip et al., 2008). Coevolution can occur at many different levels of biology: from populations and species, to adaptation of a predator to its (adapted) prey, to the evolution of a parasite/symbiont and its host, down to related mutations in amino-acids and proteins within a single organism. All members taking part in coevolution exert mutual selective pressure on each other, influencing the evolutionary process of the other. When taking place within a single biological entity, coevolution is beneficial to the entire organism. Multiple traits coevolve in order to produce individuals with a higher degree of “fitness” with respect to their environment. Cellular automata (CAs) have been used for years as a proxies for the simulation of rudimentary organisms and biological processes. In a prominent study, Mitchell *et al.* have successfully used genetic algorithms (GAs) to artificially evolve a single feature, the update function shared by all cells, of small radius one-dimensional CAs (Mitchell et al., 1993) to perform a prototypical task. However, tasks must not be only prototypical. CAs using GA

evolved functions have proven able to undertake complex tasks, applied, for instance to identifying combinations of genetic markers associated with clinical endpoints (Moore and Hahn, 2002b,a). More recently, we have conducted a study evolving a different property of CAs, the underlying network topology of CAs, with comparable success (Tomassini et al., 2005). The resulting evolved topologies are general graphs, which exhibit attributes of social network. A pioneering work by Sipper and Ruppert studied the coevolution of cellular machines (non-uniform CAs), now commonly known as (random) Boolean networks, which are non-uniform variants of the CA in which each cell has its own update function, instead of a single function shared by all cells (Sipper and Ruppert, 1997).

In this work, we propose a new framework for CA evolution consisting of the simultaneous evolution of the single update function shared by all cells (uniform CAs) and the supporting network topology of the CAs. We hypothesize that evolutionary algorithms (EAs) will generate individuals with a high capacity to solve the task at hand, and develop network topologies supporting speed, robustness, and resilience to transient failures better than that of strictly regular CAs (Tomassini et al., 2007). We compare the fitness-based performance of entire populations of the two single-feature artificial evolutions against a population of CAs simultaneously evolving both the update function and the layout of the cellular connections. Additionally, we analyze the scalability of both the existing frameworks and of the new one, as the performance of CAs with a relatively small fixed number of neighbors generally decreases with a larger number of cells. Finally, we conduct a statistical profiling of the artificially evolved network topologies in order to study the emergent properties of CAs with a higher performance.

## Background

### CAs and the Density Classification Task

CAs are dynamical, usually deterministic, discrete, abstract models used to simulate and study distributed computation. A standard CA consists of a finite number  $N$  of identical cells. Each cell can take one of a finite number of states  $s$ ,

here, the two Boolean states  $s \in \{0, 1\}$ . Each cell has a local knowledge of the state of a fixed number of  $n$  neighboring cells, including itself. The state of each cell is updated synchronously in discrete time steps, according to a local, identical update function or rule (these terms will be used interchangeably throughout this work) shared by all cells. Cells are usually arranged on a  $d$ -dimensional grid, where typically  $d \in \{1, 2, 3\}$ . In this study, we focus on one-dimensional, or linear CA, in which cells are arranged on a regular ring structure, connecting to a radius of  $r$  cells on each side. Thus the neighborhood size is  $n = 2r + 1$ . At any given *discrete time step*  $t$ , the set of all states  $s_i^t$  of all cells is called the *configuration* of the CA such that  $c^t = (s_0^t, s_1^t, \dots, s_{N-1}^t)$ , thus CAs with  $N$  nodes have exactly  $2^N$  possible configurations. Starting from an initial configuration (IC or  $c^0$ ) at time  $t = 0$ , the CA will travel across transient configurations before reaching a previously visited state of the system. After at most  $2^N$  time steps, the CA will start cycling deterministically through a subset of configurations.

**The Density Classification Task** The density classification task is a prototypical distributed computational task for CAs and is defined as follows. Let  $\rho_0$  be the fraction of 1s in the IC (i.e. time step 0). The CA's task is to determine whether  $\rho_0$  is greater than or less than  $1/2$ . If  $\rho_0 > 1/2$ , then the goal is to have the CA converge to a fixed-point configuration of all 1s; otherwise to a fixed-point configuration of all 0s, after a number of time steps with the order of  $N$ , where the CA has a odd size  $N$  to eliminate the case  $\rho_0 = 0.5$ . This computation is trivial for a computer having a central control and will provide the answer in  $O(N)$  time. However, it is nontrivial for one-dimensional CA, with a small radius, since such a CA can only transfer information at finite speed relying on local information exclusively, while density is a global property of the configuration of states (Mitchell et al., 1993).

## Graph Properties

A CA can be seen as a mathematical object known as a graph, where each cell resides on a vertex, and edges between vertices represent two neighboring cells. Therefore, formal definitions of graph theory do apply to CAs. For ease of reference, we summarize concepts used in subsequent sections particular to this work (see (Newman, 2010) for complete reference). In this work, a graph  $G$ , or network, consists of a set of  $v$  vertices  $V$ , and a set of  $e$  undirected, unweighted edges  $E$ . The degree  $k$  of a vertex is the number of edges connected to it. Thus the average degree  $\bar{k}$  of  $G$  is the average of the degree over  $V$ . A path between vertices  $u$  and  $v$  is defined as the sequence of unique edges traversed when going from  $u$  to  $v$ . Its length is the number of edges in the sequence. The average path length (APL) of  $G$  is the average length of the *shortest* path between all pairs of vertices.

The clustering coefficient  $C_j$  of a vertex  $j$  is defined as the ratio between the  $E_j$  edges that actually exist between the  $k_j$  neighbors of  $j$  and the number of possible edges between these nodes:  $C_j = 2E_j / (k_j(k_j - 1))$ . The clustering coefficient (CC) of a graph is defined as the average  $C_j$  across all vertices. The degree distribution  $P(k)$  of a graph  $G$  is a function that gives the probability that a randomly selected vertex has  $k$  edges incident to it.

## Artificial Evolution of CAs

It has been shown that the density task cannot be solved perfectly by a uniform, two-state CA with finite radius  $r < (N - 1)/2$  (Land and Belew, 1995). Despite the lack of a perfect solution, it is desirable to find one or more solutions that achieve the highest degree of performance possible.

**Evolving the Update Function** In general, it is extremely difficult to infer the local CA function that will give rise to the desired global computation due to possible nonlinearities and large-scale collective effects. On the other hand, exhaustive evaluation of all  $2^{2^n}$  possible functions is limited to small radii  $r \in \{1, 2\}$  by the computational cost. As first proposed by Mitchell *et al.* (Mitchell et al., 1994, 1993) for uniform CAs and by Sipper for nonuniform ones (Sipper, 1997; Sipper and Ruppin, 1997), EAs have proven to be a very effective heuristic to search in the colossal solution space of all update functions. Additionally, EAs have been applied to the discovery of efficient update functions for complex CA systems, such as CAs with multidimensional structure (Breukelaar and Bäck, 2005), as opposed to the linear, monodimensional nature of Mitchell's CAs, and those of interest in the present work.

**Evolving the Topology** In order to modify the underlying topology of the network supporting the cells, we will consider an extension of the concept of CAs. Therefore, cells can be connected in any way, provided that multiple edges are disallowed. Sipper and Ruppin have already examined the influence of different connectivity patterns on the density task. They studied the coevolution of network architectures and CA rules, resulting in non-uniform, high-performance networks (Sipper and Ruppin, 1997). More recently, Watts also moved away from regular structures, and hand constructed general uniform CAs for the density task (Watts, 1999). Because of the heterogeneous degree distribution of his networks, he had to reduce the update function to its simplest, and most flexible form, the majority rule (MR), to accommodate cells with varying neighborhood sizes. At each time step, each cell will assume the state of the majority of its neighbors in the graph. Watts built many networks with performance values exceeding that of CAs with evolved rules on regular lattices with similar average degree. Network structures yielding a good performance tend to have "long" links, creating shortcuts between distant

cells that somewhat compensate for the lack of information transmission of the regular lattice case.

In both these studies, the authors correctly recognized that reducing the average cell to cell distance, i.e. the APL, has a positive effect on the performance of the CA. Inspired by their work, we have explored the effect of artificially evolving the underlying topology of CAs with uniform MR starting with a population of regular lattice and one of random topology CAs (Tomassini et al., 2004, 2005, 2007). Evolved networks resulting from either initial populations tend to converge in the “small-world” region of the spectrum between regular structures and random ones (Watts and Strogatz, 1998). Indeed, these evolved topologies exhibited long reaching shortcuts across the network, thus significantly shortening the APL. Their CC is higher than that expected of equivalent random networks. Finally, their degree distribution was slightly skewed to the right, showing the emergence of a few higher-than-average connected hub cells, which is another property of social networks.

## Methods

In order to compare and contrast the performance of existing paradigms to that of our coevolution of topology and rules, we have implemented the two single-feature evolutions according to the original framework specifications. However, in order to make them comparable to each other and to our work, some parameters needed to be harmonized. Unless otherwise specified, parameters are identical across all simulation sets.

### Evolutionary Algorithm for CAs

In this work, we use an EA with the aim of evolving CAs for the density task. We assume the reader is familiar with the concept of artificial evolution, evolutionary computation, evolutionary algorithms, and genetic algorithms (Holland, 1975; Bäck, 1996; Mitchell, 1996). For all 100 experimental replicates, we generate an initial unstructured population of size  $P = 100$  individuals (i.e. individuals are not spatially limited in their interactions). The definition of individuals varies according to the framework used: update function, network topology, or both simultaneously (see below). We explore the scalability of the systems by studying CAs of size  $N \in \{99, 199, 299\}$ . Regardless of the size  $N$  and of the framework used, the initial population at generation  $g = 0$  is made of  $P$  uniform regular (i.e. ring) CAs with a radius  $r = 3$ , thus the neighborhood size of each cell is  $n = 7$ , including itself. The radius changes as the EA progresses when evolving the topology and with coevolution. Evolution is ended when the entire population has reached an optimal fitness, or after a maximum of  $g_m$  generations. We find experimentally that  $g_m = 100$  is enough for the populations to reach a *fitness plateau* where improvement becomes marginal or null.

The fitness function of the EA used to evaluate the “quality” of a CA individual in the population consists of experimentally evaluating the ability of the CA to solve the density task over a sample of 100 ICs with uniformly distributed densities  $\rho \in [0, 1]$ . CAs are allowed a maximum number of  $2N$  time steps to converge. If two consecutive configurations  $c^t$  and  $c^{t+1}$  are identical, the CA is stopped, as it has reached a *single configuration* attractor, and all consecutive configurations will remain identical. The fitness is defined as the fraction of instances (i.e. ICs) for which the CA produces the correct fixed point, given the known density of the IC. At each generation a different set of ICs is generated for each individual.

The next generation is obtained by repeating  $P$  standard *binary tournament* selection over the entire population. We describe the mutation methods below, as they depend on the evolution strategy. However, the mutation rates are selected quantitatively to yield the best results. The concept of *recombination* of network individuals is ill defined, and cumbersome to implement. However, we have used standard single-point recombination in frameworks that evolve the update function with a probability  $p_c = 0.25$ . At the end of each evolutionary process, we select the *elite population* (EP) by selecting the CAs that fall in the 95<sup>th</sup> percentile of *performance* (i.e. all individuals with a performance that is within the absolute best). For performance evaluation, the entire population is exposed to 1,000 instances of the most difficult problems possible, that is, on ICs with  $\rho \approx 0.5$ . Note the difference between *fitness* (100 uniform ICs) and *performance* (1'000 difficult ICs). In order to obtain statistically sound results, we replicate all experiments 100 times, and record the average fitness, the best fitness at each generation, and network statistics, such as the degree distribution, the APL and the average CC at the end of the evolutionary process.

**Evolution of the Update Function (UFE)** When we evolve the update function of uniform CAs with fixed topology, each individual of the initial population is assigned a different random update function in the form of a Boolean lookup table. This table is shared by all cells in the CA and its size is  $2^n$ , where  $n = 7$  is the size of the neighborhood. At each time step, cells synchronously update their binary state according to the state of their neighborhood. The ordering of the neighborhood is predetermined, thus making the system fully deterministic. At every generation, selected parent individuals will produce mutated offspring. The mutation will impact the offspring’s lookup table and is susceptible to change the binary value of each position in the table with a probability  $p_m = 5 \times 10^{-3}$ . The offspring will replace its weaker parent, if it has a better fitness than either. This approach is similar, although not entirely identical to that of Mitchell *et al.* (Mitchell et al., 1994, 1993).



**Evolution of the Topology (TE)** When evolving the topology, each regular uniform CA of the initial population is assigned the same update function, that is, the MR. At each time step, cells will update their state to reflect the majority of its neighborhood. As mentioned previously, the MR has been proven incapable of solving the density task when applied to a regular CA. In order to increase its fitness, we allow each CA to modify the structure of its supporting network. At every generation, selected individuals produce a mutated offspring that has a high probability of replacing its parent in the next generation if it shows a higher degree of fitness. Each cell will see its neighborhood mutated with a probability  $p_m = 5 \times 10^{-3}$ . If a cell is mutated, it will lose one of its neighbor or gain a random one with equal probability (Tomassini et al., 2005). This operator prevents drastic changes in the average degree of the CAs, which would giving an advantage to those with a higher connectivity (i.e. more edges). We disallow mutations that would produce self-loops or duplicate edges. In the case of a tie in an even sized neighborhood, the state will be drawn at random.

**Coevolution (CE)** This is the new framework we propose, allowing simultaneous evolution of the update function and the underlying topology of the network. Similar to the evolution of the update function only, coevolution starts with a population of uniform regular CAs, each with a randomly generated lookup table. As in the precedent frameworks, selected parents at each generation will produce mutated offspring, which may replace its parents in the subsequent generation. Mutations will now affect first the topology of the CA, by mutating edges with a probability  $p_t = 0.001$ , and then the lookup table, with probability  $p_r = 0.003$ . In this case, the size of the lookup table might need to be adapted to the growing sizes of the neighborhoods. Indeed, every time the size  $n_{max}$  of the largest neighborhood increases by 1, where  $n'_{max} = n_{max} + 1$ , the table size doubles, growing from  $2^{n_{max}}$  to  $2^{n_{max}+1} = 2^{n'_{max}}$ . The new half of the lookup table is completed with randomly selected values with equal probability. This ensures that, when the new neighbor is *off*, the target cell behaves just as it did before the new neighbor was added. When it is *on*, the target cell has a whole new behavior.

## Experimental Results

In order to compare results with previously proposed evolutionary CAs frameworks, we conducted parallel simulations for all possible combinations of frameworks (rule-only, topology-only, and coevolution) and sizes  $N \in \{99, 199, 299\}$ . At each generation of the artificial evolution, we record the average fitness of each population, and the fitness of the best individual. After the last generation has been produced, we segregate the EP as described above. We compute standard graph statistics on the CA networks of EP to shed some light on the mathematical properties that

set the evolutionary frameworks apart from one another.

## Performance and Fitness

The ultimate goal of evolving CA for computation, regardless of the framework, is to obtain individuals that excel at solving both the average case and the “worst case scenario” of the task at hand. In fact, a single individual is enough as long as its performance is satisfying the task’s criteria of quality and speed. Figure 1 shows the results of the performance evaluation of the EP on difficult problems. Each column represents a combination of frameworks and CA sizes. Performance, just like fitness, is on a scale from  $[0, 1]$  representing the fraction of correctly classified IC in at most  $2N$  time steps. We show the consistency of the EP results by showing the absolute best performance (upper mark), the average performance (horizontal line), and the lowest performance (lower mark). The number in parentheses is the size of the EP for each case, and gives an ideas of how rich the solution space is in “good individuals”.

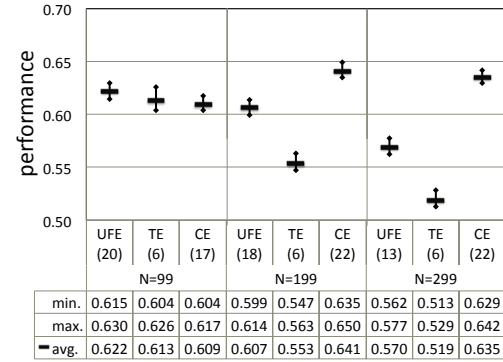


Figure 1: Performance of Best Evolved CAs. Performance of all three frameworks: update function/rule evolution (UFE), topology evolution (TE), and coevolution (CE) of individuals  $N \in 99, 199, 299$  in the elite population (EP). Performance is measured over 1,000 ICs with  $\rho \approx 0.5$ . Average number of individuals in each run is in parentheses. In each column, the horizontal bar represents the average performance, the bottom mark represents the lowest performance, and the top mark shows the best performance. Results are averaged over 100 independent simulations.

In the case of smaller CAs, results across all frameworks are virtually undistinguishable, yet consistently above the 0.6 mark. In any case considered, the deviation between the maximum and the minimum fitness is minute ( $\sim 0.02$ ). The clear distinction comes with the scaling of the systems to larger CAs. The performance of TE CAs drops significantly as the system grows. Similarly, the UFE suffers a decrease of performance as the system grows, though not as much as TE CAs. On the other hand, coevolving CE systems see their performance remain stable or marginally improve as

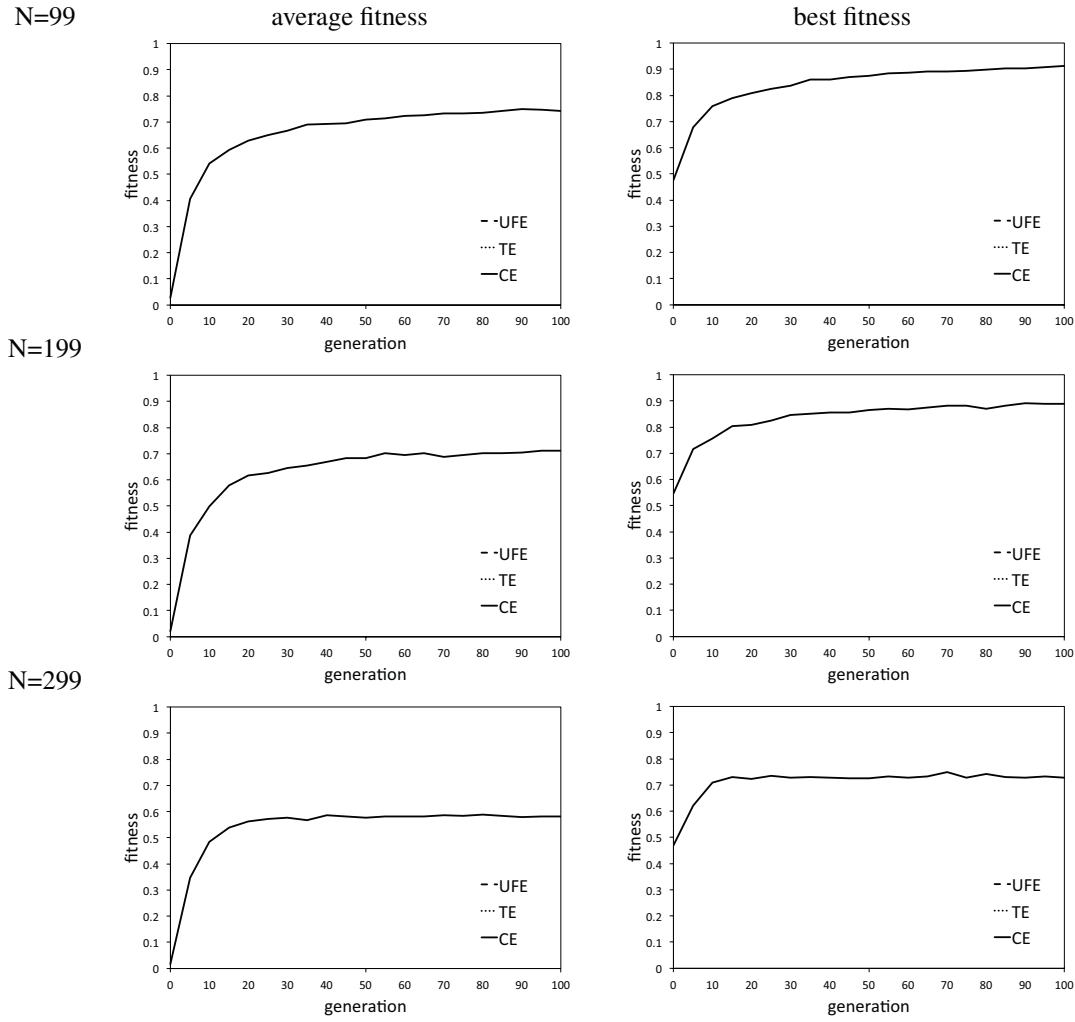


Figure 2: Fitness progression of Evolutionary CAs. Average fitness over the course of 100 generations for populations of evolutionary CAs (left column) and fitness of the best individual (right column). Systems of size  $N = 99$  (top row),  $N = 199$  (middle row), and  $N = 299$  (bottom row). Each panel shows results of all three frameworks: rule evolution (UFE, dashed line), topology evolution (TE, dotted line), and coevolution (CE, continuous line). Results are averaged over 100 independent simulations.

the CAs get larger. Interestingly, UFE and CE seem to consistently evolve a larger number of good individuals (Figure 1, numbers in parenthesis) than TE, therefore the solution space of TE is likely to be the poorest in EP CAs. This difference can be explained by the complexity and overall size of the actual solution space. In the case of UFE, there is at most  $2^{2^n}$  different update possible that constitutes the entire space. In the case of TE, however, the solution space is much larger, made of all possible topological network structures of  $N$  vertices. Surprisingly, the EP size of CE is also large, despite of the much larger solution space, made of all possible topologies coupled with all possible update function.

To understand the results presented above, we analyze the time progression of our evolutionary CAs over the 100 generations. We trace the development of UFE, TE, and CE

populations' average fitnesses in the left-hand column of Figure 2, whereas the right-hand side panels show the curves for the fitness of the best individual in the CA population, with the highest fitness value. Additionally, we show the scaling of these systems, which will also help us appreciate the shape and richness (or lack thereof) of the different solution spaces. When comparing the different combinations of evolutionary framework and sizes, we notice that the general trends are similar across all panels of a column in Figure 2. CAs under UFE and CE have the steepest learning curve. CE yield the second steepest curve, reaching a lower, CA size dependent, fitness plateau almost at the same time as UFE. The difference of average fitness between CE and UFE increases notably with the scaling in size  $N$ . However, we notice that this difference is less pronounced in the best

fitness curves in the right panels. This might explain the results in performance presented in Figure 1. The trace of TE has a different shape, with a slow start, turning into a steep slope, only reaching its fitness plateau consistently significantly higher than that of UFE or CE. Surprisingly, only the average population of TE seems impacted by the scaling. Additionally, in the average fitness panels, we note that the UFE and CE curves start at a fitness close to 0, where as TE starts with a lead of about 0.5, we note that the final fitness of TE is notably above that of UFE and CE, which seems to contradict the performance results above. In fact, we see the importance of the probability  $\rho$  of the ICs. Indeed, topology-only evolutionary CAs perform on average better than the other two on easier problems.

Best fitness results in the right-hand side panels of Figure 2 are consistent with those of the average fitness. Again, TE CAs reach a near optimum fitness, despite a slower start. The best individuals under UFE and CE preform closely, although UFE become clearly superior to CE with the increasing size of the system. Which again proves that the solution landscape of these EAs is not necessarily favorable to those rich in good individuals, as even randomly initialized individuals are capable of high fitness. Moreover, the fact that topology-only evolution yield a best fitness near 0.5 at the first generation also agrees with Mitchell's finding that uniform topologies with the majority rule cannot solve the density task better than a random system. The scaling of the CA sizes impacts the fitness of the coevolution framework the most, both in average and maximum fitness. We even witness a slight drop in best fitness for  $N = 199$ , a drop more prominent in  $N = 299$ , where after a slight, or no increase at all, the best fitness plateaus at a value lower than that of the maximum reached earlier in the evolution. We can only explain the fact that CE is detrimental for larger systems' fitness by the competing evolution of the two traits simultaneously. Nevertheless, the artificial evolution is beneficial in terms of producing CAs capable of consistently solving the task with any density  $\rho$ , especially those closest to  $\rho \approx 0.5$  and is clearest in the larger CAs. The results of UFE and TE are much more sensitive to the difficulty of the problem, where the fitness drops in both cases below the performance of CE in larger systems under difficult problems.

In summary, we note that the fitness in CE's may be less associated with performance than in the other models. This may indicate that high fitness does not (linearly) translate into high performance. It would appear that the TE's are the most flexible at solving the majority problem. While they may not be the best on the really hard problems, they show great adaptability to the a wide range of initial conditions.

### Properties of Evolved Topologies

EA's only goal is to optimize the performance of the CAs. However, looking beyond the fitness and performance, we are interested in studying the properties emerging from

the evolved topologies, and how they differ when obtained solely by topology evolution, and when the network's evolution is combined with adaptations of the update function. Therefore, we analyze the graph and statistical properties of the supporting network structure of CAs before the evolutionary process starts (i.e. a regular ring structure) and after the last generation of the EA. Figure 3 offers a visual representation of sample CA structures before (A) and after the evolution (B) and (C). Figure 3(A) shows the regular topol-

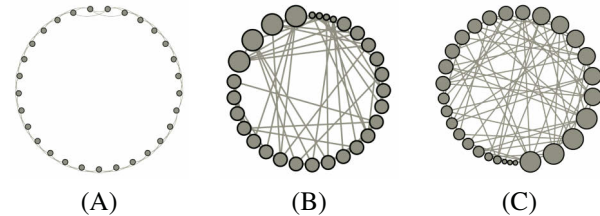


Figure 3: CA Topologies. Instances of a regular CA's initial network topology (A), and the CA topologies resulting from topology-only evolution (B) and coevolution (C). In the bottom row, the size of the vertices is proportional to the size of the neighborhood of the cell (i.e. degree of the cell). CA size  $N = 29$  for ease of reading.

ogy of UFE CAs (at all times) and of all CAs before the specific evolutionary process takes place. All vertices have a degree  $k = 4$ , and are thus all the same size. In the example graphs post-evolution (B) and (C), we see the emergence of cells of higher and lower degrees. The average degree of the networks is, however, maintained at all times  $\bar{k} \approx 4$  by the edge mutation within the graph (see Methods' section). The vertices' sizes are proportional to their degree, and ordered according to that criteria. At first glance, TE gives rise to a greater diversity in the vertices' degrees, whereas CE CAs seem more homogeneous in their degree distribution. The degree distributions of both frameworks, topology evolution and coevolution, and sizes of CAs are depicted in Figure 4. The values for the degree distributions are averaged over the elite population of the 100 replicates. From the degree distributions in Figure 4, we can see the EA has shifted the peak of the CE from all nodes having a degree  $k = 7$  to a majority of  $k = 8$ . Although there is some spread in the degrees, the function is narrowly centered around its peak, with little deviation, and no extreme values. TE has, on the other hand, facilitated a larger heterogeneity in the degree distribution, with a significantly wider bell shaped curve, and no clear peak at a single value of the degree  $k$ . TE CA networks has therefore more extreme values, where the minimum degree of a TE network is smaller than that of a CE network of the same size:  $k_{min}^{TE} < k_{min}^{CE}$ . The opposite is true of maximum degrees:  $k_{max}^{TE} > k_{max}^{CE}$ . The number of cells does not appear to have a marked influence on the shape of the degree distribution of emerging networks. The size of the CA only affects the magnitude of the function, not the shape. In Table 1, we report two essential statisti-

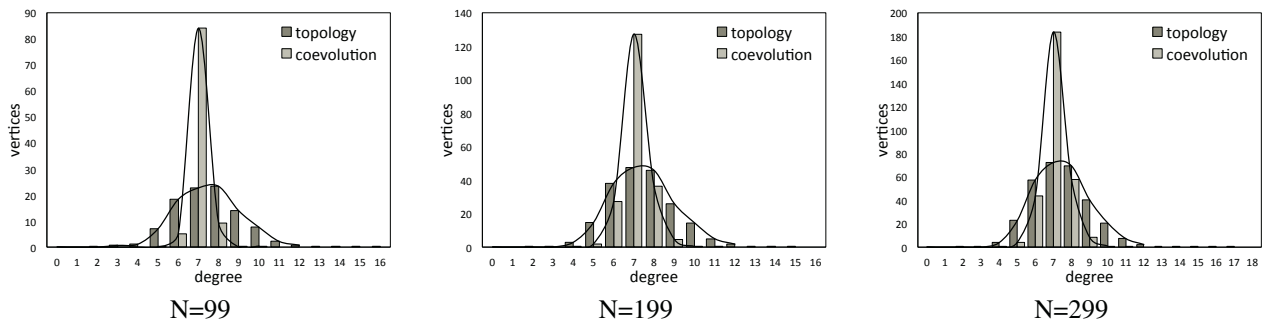


Figure 4: Degree Distributions of Evolved Topologies. CA topologies of sizes  $N \in 99, 199, 299$ . The degree distributions (bar plots) show the number of vertices (Y-axis) having a given degree (X-axis). Each panel contains the results for CAs with evolution of the topology-only (dark grey) and coevolution (light grey). Results are averaged over 100 independent simulations. Continuous lines are only meant as a guide for the eye to the trend of the degree distribution probability functions.

cal properties to the study of the CA topologies, the average path length and the average clustering coefficient after the evolution. Formal definitions of these two metrics can be found in the Background section.

The degree distributions in Figure 4 and the values of APL and CC in Table 1 suggest that the network structures emerging from artificial evolution share properties with technological, social, and other “real-world” networks. These architectures generally show greater resilience to perturbations than regular structures (Newman, 2010; Watts, 1999). Moreover, the even degree distributions, with a few hub-like cells, the short APL and higher CC of graphs after coevolution are all properties placing them even closer to “real-life” on the spectrum of all graphs.

### Speed of Convergence

If the quality of the results is key to the success of a CA, the speed at which the CA will converge to a solution is a non-negligible factor. In our previous studies (Tomassini et al., 2005, 2007), we show that the diffusion of information is immensely facilitated by the emergence of shortcuts across the networks, thus the shortening of the APL. Due to limited space, we present in Figure 5 only three examples representative of the ability of evolved-topology CAs to converge to a solution faster than regular CA, regardless of the excellence of their evolved rules.

In Figure 5, we show the progression of evolved CAs through the configuration space converging towards the correct IC density value. Mitchell *et al.* have analyzed the emerging patterns visible in this type of figure (Mitchell et al., 1993). In our results, TE CAs are consistently orders of magnitude faster than CE (over 500%), which are, in turn, significantly faster than UFE (10 – 300%) to reach a steady configuration.

### Conclusions & Future Work

Cellular Automata are, despite their apparent simplicity, powerful models for distributed computations, provided that

update function evolution (UFE)      topology evolution (TE)      coevolution (CE)



ATS	UFE	TE	CE
N=99	90.37	12.41	76.15
N=199	141.08	18.16	127.07
N=299	440.20	23.32	132.21

Figure 5: Density Classification Task performed by Evolved CAs. Examples of single time step (vertical axis) evolutions of evolved CAs solving the density classification task starting from an arbitrary IC with  $\rho \approx 0.5$  (top row of the horizontal axis in each panel) for all three CA evolution frameworks. The table below the panels show the average number of time steps (ATS) necessary to successfully solve the density task over 1,000 ICs.

an adequate update function can be found. In the present work, we demonstrate once again the ability of EA to develop highly performant rules to solve a prototypical CA task, and remaining unaffected by the scaling of the CA size. The solution space is, in this case, rich in good solutions, making the EA capable of finding large numbers of functions that perform well. From a distributed computation perspective, the regular topology of standard CAs can be a weakness, as these structures are more susceptible to fail under transient perturbation. One alternative is to evolve the topology, and leave the majority rule as a constant update function. This TE paradigm evolves topologies showing properties of resilient systems, and the resulting CAs are a great deal faster than UFE. Unfortunately, their



Table 1: Evolved CAs Network Statistics. Average path length (APL) and average clustering coefficient (CC) for evolved CA networks in the EP for each evolutionary framework, and CA sizes  $N \in 99, 199, 299$ .

	N=99			N=199			N=299		
	rules	topology	coevolution	rules	topology	coevolution	rules	topology	coevolution
APL	8.59	2.53	4.410	16.91	2.97	5.11	25.25	3.23	4.68
CC	0.71	0.21	0.67	0.71	0.21	0.67	0.71	0.22	0.59

performance is inversely proportional to the density, and to the size of the CA. In this work, we propose a framework that simultaneously evolves the update function and topology underlying. We have developed a novel update function implementation that integrates with the changing topology. The CAs resulting from CE demonstrate performance levels comparable to, and scale better than UFE, and are considerably faster. Moreover, the performance of CE systems remains constant (even slightly increases) as we scale up the size of the CAs. On the contrary, this scaling affects adversely the performance of both UFE and TE. Finally, they exhibit properties possibly making them even more robust network systems than TE CAs. We are planning on implementing a complementary study for structured populations, as they have been shown to increase the EA performance. In addition, we will conduct a thorough analysis of the resilience and fault tolerance of all evolved CAs (Tomassini et al., 2005).

## Acknowledgments

The authors gratefully acknowledge Britney E. Graham at Dartmouth College for her help in editing this article. CD and JHM gratefully acknowledge the financial support of the NIH (grants AI59694, LM010098 and LM009012). MG gratefully acknowledges the financial support of the local research founding program of the University of Torino.

## References

- Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, New York.
- Breukelaar, R. and Bäck, T. (2005). Using a genetic algorithm to evolve behavior in multi dimensional cellular automata: emergence of behavior. In *GECCO*, pages 107–114.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, Michigan.
- Land, M. and Belew, R. K. (1995). No perfect two-state cellular automata for density classification exists. *Physical Review Letters*, 74(25):5148–5150.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA.
- Mitchell, M., Crutchfield, J. P., and Hraber, P. T. (1994). Evolving cellular automata to perform computations: mechanisms and impediments. *Physica D*, 75:361–391.
- Mitchell, M., Hraber, P. T., and Crutchfield, J. P. (1993). Revisiting the edge of chaos: evolving cellular automata to perform computations. *Complex Systems*, 7:89–130.
- Moore, J. H. and Hahn, L. W. (2002a). Cellular automata and genetic algorithms for parallel problem solving in human genetics. In *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, PPSN VII, pages 821–830, London, UK, UK. Springer-Verlag.
- Moore, J. H. and Hahn, L. W. (2002b). A cellular automata approach to detecting interactions among single-nucleotide polymorphisms in complex multifactorial diseases. In *Pacific Symposium on Biocomputing 2002*, pages 53–64. <http://www.odysci.com/article/1010112988060408>.
- Newman, M. (2010). *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA.
- Sipper, M. (1997). *Evolution of Parallel Cellular Machines: The Cellular Programming Approach*, volume 1194 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, New York.
- Sipper, M. and Ruppín, E. (1997). Co-evolving architectures for cellular machines. *Physica D*, 99:428–441.
- Tomassini, M., Giacobini, M., and Darabos, C. (2004). Evolution of small-world networks of automata for computation. In et al., X. Y., editor, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 672–681. Springer Verlag, Berlin.
- Tomassini, M., Giacobini, M., and Darabos, C. (2005). Evolution and dynamics of small-world cellular automata. *Complex Systems*, 15:261–284.
- Tomassini, M., Giacobini, M., and Darabos, C. (2007). Performance and robustness of cellular automata computation on irregular networks. *Advances in Complex Systems*, 10:85–110.
- Watts, D. J. (1999). *Small Worlds: The Dynamics of Networks Between Order and Randomness*. Princeton University Press, Princeton, NJ.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of "small-world" networks. *Nature*, 393:440–442.
- Yip, K. Y., Patel, P., Kim, P. M., Engelman, D. M., McDermott, D., and Gerstein, M. (2008). An integrated system for studying residue coevolution in proteins. *Bioinformatics*, 24(2):290–292.

# Bipartite Networks Show the Genotype-to-Phenotype Relationship in Biological Systems Models:

## A Study of the Robustness, Evolvability, and Accessibility in Linear Cellular Automata

Christian Darabos<sup>†</sup>, Britney E. Graham, Ting Hu, and Jason H. Moore

The Geisel School of Medicine at Dartmouth College, Hanover, NH 03755, USA

<sup>†</sup>Christian.Darabos@dartmouth.edu

### Abstract

In biological organisms, a single genotype may map to several phenotypes and vice-versa. This many-to-many relationship is believed to be a major drive of the phenotypic robustness and genotypic evolvability found in all life forms. Given the inherent complexity of the genotype-to-phenotype (G2P) mappings, we use cellular automata (CAs) as rudimentary proxies for biological organisms. CA models have the same many-to-many G2P mappings, and their sensitivity to initial conditions allows the same genotype to differentiate into different phenotypes. We use a bipartite network to study the G2P landscape, and its projections in either space. The degree distributions of the network and its projections are all *heavy-tailed*, denoting the presence of highly connected hubs, implying that increased robustness is supported by the network structure. We also show a strong correlation between the phenotype's complexity and its robustness. We analyze the relationships between the robustness and the evolvability both at the genotypic and phenotypic level. Although we use different computational models, our results agree with those of previous similar studies, and with observations in biological organisms.

### Introduction

For the past two decades, geneticists have been studying the intricate genotype-to-phenotype (G2P) relationship in biological organisms. Genome-wide association studies (GWAS), and the recent advances in modern high throughput sequencing technologies, have made understanding how metabolic reactions, cell signaling, and developmental pathways translate the genome of a living organism into its phenotype an achievable goal (Nuzhdin et al., 2012). However, GWAS have also unveiled unprecedented degrees of complexity, making clinical progress much slower than anticipated. As geneticists learn more about G2P mappings, it becomes more apparent that there is a many-to-many relationship. Indeed, several different genotypes, usually resulting from small perturbations or *neutral mutations*, result in the exact same phenotype. This feature is responsible for the phenotypic robustness of biological organisms, and their relative insensitivity to small genetic perturbations. On the other hand, identical genotypes may develop into dramatically different phenotypes, depending on a set of internal

and external signals and factors. The embryonic stem cell, which may potentially develop into any cell type, is a prime example of a single genotype yielding several phenotypes. The ability to adapt to internal and external factors is believed to be a major factor of the evolvability of all life forms. Given the inherent complexity of the G2P mappings, we recognize the need for smart, adaptive mathematical, statistical or computational models to study this relationship.

In this work, we use a cellular automata (CAs) model to exhaustively explore the G2P relationship in basic models of biological organisms. We are specifically interested in the nature of CAs, which in their simpler form mimic the many-to-many G2P relationship, and their sensitivity to initial conditions other model systems fail to encapsulate. CAs have been thoroughly studied in the past. However, in the context of this project, we will focus on the exhaustive description of all possible genotypes, phenotypes, and initial conditions. The results we gain from this are structured in a bipartite network, which consists of two types of nodes, in our case: genotype and phenotype. We then look at the projections of our bipartite network onto the genotype space and the phenotype space. Additionally, we study the distribution of robustness (also called neutrality) (Banzhaf et al., 1994) in the genotypic landscape of our model, and its effect on the phenotypic landscape. Similarly, we look at the genotypic and phenotypic evolvability, and the correlations between robustness and evolvability. Indeed, the seemingly contradictory effect of robustness and evolvability has been studied and disproved in many systems, where they in fact facilitate each other (Bloom et al., 2006; Ferrada and Wagner, 2008; Wagner, 2008a, 2005).

### Background

#### Cellular Automata (CAs)

CAs (Codd, 1968) are dynamical, usually deterministic, discrete, abstract models used to simulate and study distributed computation. A standard CA consists of a finite number  $N$  of identical cells. Each cell can take one of a finite number of states  $s$ , here, the two Boolean states  $s \in \{0, 1\}$ . Each cell has a local knowledge of the state of a fixed num-

ber of  $n$  neighboring cells, including itself. The state of each cell is updated synchronously in discrete time steps, according to a local, identical update function or rule (these terms will be used interchangeably throughout this work). In all CAs, update functions are generally represented as a Boolean lookup table of all possible binary permutations of the cell's neighborhood. Cells are usually arranged on a  $d$ -dimensional grid, where usually  $d \in \{1, 2, 3\}$ . In the present case of a one-dimensional, or linear CAs, cells are arranged on a regular ring structure, connecting to a radius of  $r$  cells on each side. Thus, the neighborhood size is  $n = 2r + 1$ . At any given *discrete time step*  $t$ , the ensemble of all states  $s_i^t$  of all cells is called the *configuration* of the CA such that  $c^t = (s_0^t, s_1^t, \dots, s_{N-1}^t)$ , thus CAs with  $N$  nodes have exactly  $2^N$  possible configurations. Starting from an initial configuration (IC or  $c^0$ ) at time  $t = 0$ , the CA will travel across transient configurations before reaching a previously visited state of the system. Because of its deterministic nature, the CA will, after at most  $2^N$  time steps, start cycling deterministically through a subset of configurations, called an *attractor*. Figure 1 shows a small example of a 8 cell regular uniform CA, set to a random IC.

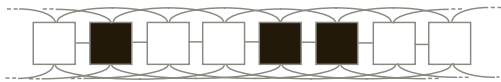


Figure 1: Regular uniform CA. Size  $N = 8$ , with a radius  $r = 3$ , and set to a random (initial) configuration. Black cells represent state 1 and white is 0. Dashed edges “wrap around” to connect the CA into a ring topology.

**CA Genotype & Phenotype** CAs have been used for years as a rudimentary proxy for biological organisms and phenomena. One prevalent example, using a generalized form of CAs, is Kauffman’s Random Boolean Network (RBN) model for genetic regulatory networks (Kauffman, 1969). RBNs use non-uniform, unstructured CAs where each cell has its own Boolean update function (BUF) and can arbitrarily be connected to any other cell. Other models, such as Genetic Programming, Bayesian systems, or differential equations have also been thoroughly studied. CAs possess, however, a modest advantage. Indeed, CAs have a genotype, a phenotype, and mimic the many-to-many G2P mappings. The update function of CAs is a direct equivalent of a genotype, which can be mutated at will, and is a set of rules followed by the system to achieve a steady state. The attractor reached by the CAs is the phenotype resulting from a genotype and an initial configuration. The same attractor can be reached by different BUFs, and a single BUF can result in different attractors depending on the IC. In this work, we explore the evolvability, robustness, and accessibility of pseudo-biological organisms modeled by a small CA. We exhaustively explore all G2P mappings of a CA by repre-

senting it in a bipartite network, and also projecting it onto the phenotype and genotype landscape respectively.

Alternative methods have been used to map the G2P relationship, such as genetic programming Hu et al. (2011), and random Boolean networks Kauffman (1969). Moreover, one can imagine using almost any systems where a genotype and a phenotype can be described, for example NK-landscapes Kauffman and Weinberger (1989); Ochoa et al. (2008), and many more. However, we determined that CAs were the best suited tools to simulate the many-to-many G2P mappings, while keeping the analytical simplicity.

## Network Properties

A CA can be seen as a mathematical object known as a graph, where each cell resides on a vertex, and edges between vertices represent two neighboring cells. Therefore, formal definitions of graph theory do apply to CAs. For ease of reference, we summarize concepts used in subsequent sections particular to this work (see (Newman, 2010) for complete reference). In this work, a graph  $G$ , or network, consists of a set of  $v$  vertices  $V$ , and a set of  $e$  undirected, unweighted edges  $E$ . The degree  $k$  of a vertex is the number of edges connected to it. Thus the average degree  $\bar{k}$  of  $G$  is the average of the degree over  $V$ . A path between vertices  $u$  and  $v$  is defined as the sequence of unique edges traversed when going from  $u$  to  $v$ . Its length is the number of edges in the sequence. The average path length (APL) of  $G$  is the average length of the *shortest* path between all pairs of vertices. The clustering coefficient  $C_j$  of a vertex  $j$  is defined as the ratio between the  $E_j$  edges that actually exist between the  $k_j$  neighbors of  $j$  and the number of possible edges between these nodes:  $C_j = 2E_j / k_j(k_j - 1)$ . The clustering coefficient (CC) of a graph is the average of  $C_j$  over  $V$ .  $C$  is thus independent of  $N$  for a regular lattice, and approaches  $3/4$  as  $k$  increases. The degree distribution  $P(k)$  of a graph  $G$  is a function that gives the probability that a randomly selected vertex has  $k$  edges incident to it. For a random graph  $P(k)$  is a binomial peaked at  $P(k)$ . But most real networks do not show this kind of behavior. In particular, in scale-free graphs which are frequent in real-life,  $P(k)$  follows a power-law distribution.

**Bipartite Network** A bipartite network consists of two disjoint sets, or types, of nodes  $U$  and  $V$ . The nodes are connected in such a way that the nodes of one set will have no connections between them, but can only be connected to nodes of the other set. The use of a bipartite network is natural when dealing with two different types of data sets (see Figure 2b). Two nodes of the same type cannot be connected to each other, so one node can only be connected to a node of the other data type. We are interested in using a bipartite network to represent the relationship of our data.

From the bipartite network, one can project the data onto either the space  $U$  or  $V$  (Figure 2a,c). In either single dataset

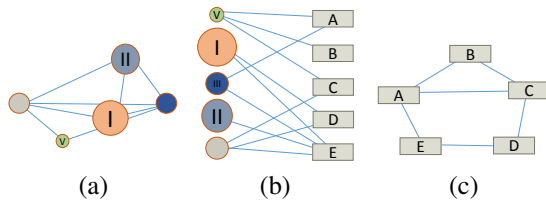


Figure 2: Bipartite Network schematic. A bipartite network (b) made of two data sets the circles,  $U$ , and the rectangles,  $V$ . Projections in  $U$  (a) and in  $V$  (c).

space, the nodes are connected to one another “through” a vertex of the the other space. By ignoring the different types of data, all network properties described above remain valid on the bipartite network (as a single data set network) and on either projection. This type of network gives us three degree distributions, one for each projection, and one for the bipartite network. Each degree distribution shows how many links each node has. Nodes in a projection of a bipartite network are connected if they share at least one node in the other group. This gives us the ability to see the interactions within a group.

## Methods

In order to fully explore the G2P relationship in our CA model, we exhaustively explore all possible genotype mappings for all possible ICs. Unfortunately, the (super)exponential nature of the genotype and phenotype spaces, we are limited to a small number of cells,  $N = 5$ , and a *radius*  $r = 1$ , where the radius defines the number of neighbors each cell arranged on a ring can reach on either side. Therefore, a radius  $r = 1$  results in a neighborhood size of  $n = 3$ . In CAs, there are  $2^{2^n} = 2^{2^3} = 256$  possible genotypes. This is the limit, as the next possible regular neighborhood size being  $n = 5$ , which results in  $\approx 4.3 \times 10^9$  genotypes. This figure is computationally too expensive to search exhaustively. CAs have  $2^N = 2^5 = 32$  possible ICs, and the same number of possible point (i.e. single configuration) attractors, and at most  $2^{2^n} \times 2^N = 8192$  possible attractors of any length, as every combination of genotype and IC can potentially result in a different phenotype. We have successfully simulated CAs of size up to  $N = 12$ , the resulting networks are however not suited for representation.

Because we have two types of data, we can build a bipartite network. In our case, one set of nodes represents genotypes and the other, phenotypes. Our network is directed and shows the descent of a phenotype from a genotype. The degree distribution of the set of genotypes gives the number of phenotypes that are associated with each genotype and vice versa.

From our bipartite network, we can build the 2 other networks, one of each set of nodes. From this, we obtain a phenotype network and a genotype network. The nodes in each

of these networks are connected if, in the bipartite network, they share at least one node of the other type. For example, in our phenotype network, two nodes are connected if they share at least on genotype. We carry out the same process for the genotype network.

## Robustness, Evolvability, and Accessibility

Several measures of robustness and evolvability exist in the literature, at both the genotypic and phenotypic scales. Following (Wagner, 2008b), we define genotypic robustness as the fraction of the total number of possible point mutations to a given genotype that are neutral. Genotypic evolvability is defined as the fraction of the total number of possible phenotypes that are accessible through non-neutral point mutations to a single genotype. Phenotypic robustness is defined as the size of the phenotype’s underlying genotype network. For phenotypic evolvability, the proportion of the total number of phenotypes that can be reached via non-neutral point mutations from a given phenotype.

In addition to measuring the propensity to mutate away from a phenotype, we also measure phenotypic accessibility (Cowperthwaite et al., 2008), denoted by  $A$ , which is formally defined as:

$$A_i = \sum_j f_{ij},$$

where

$$f_{ij} = \begin{cases} \frac{v_{ij}}{\sum_{k \neq i} v_{ik}}, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases}$$

This metric takes on high values if phenotype  $i$  is relatively easy to access from other phenotypes, and low values otherwise.

## Results

This section describes the analyses resulting from the exhaustive simulation of the G2P relationship under all possible initial conditions in a small CA. We begin by describing the G2P spaces in the form of its bipartite network and its projections, and studying the networks’ statistical properties. The second part of this section focuses on the correlations between the genotypic and phenotypic robustness, evolvability, and accessibility.

### Bipartite Network of G2P

We start by representing the G2P relationship in our CA model as a bipartite network. Figure 3 represents this bipartite network, the degree distributions, and both projections. Figure 3a shows the bipartite network, where genotypes (on top) only connect to phenotypes (on the bottom). To the best of our knowledge, this is a novel perspective on representing the G2P mappings. In all network Figures 3a-c-d, the size of the vertex is proportional to the number of members of



the opposite dataset associated. In other words, the genotypes vertices are proportional to the number of phenotypes mapped, and vice-versa. For readability reasons, we have filtered out vertices of a degree below 5. In the same figure, we also show the trends of degree distributions for the bipartite network, as well as both projections of the genotypes only and of the phenotypes only.

We identify the two most evolved phenotypes as the two single-point homogeneous phenotypes (00000 and 11111). The next most evolved phenotype is a two-state attractor made from the previous two. As expected, the identity update function (01010101) is the genotype that maps to the most phenotypes. In Figure 3b, all degree distributions are right skewed, with a heavy tail, denoting the presence of highly connected “hub” vertices and a “scale-free” like topology. In other words, the degree distribution of the network decays like a power-law or exponential function.

Beyond the degree distributions, we are also interested in showing a few more statistical characteristics of the networks, and how they differ between the bipartite network and the projections. Table 1 summarizes these measurements, described in the Methods Section.

network	bipartite	genotype	phenotype
#vertices	395	256	139
#edges	1398	1024	1268
$\bar{k}$	3.539	8.0	18.245
CC	0	0	0.816
APL	1	4.016	2.102

Table 1: Networks statistics for the bipartite network as a whole, and both the projection on the genotypes space and on the phenotypes space.

As we can see in Table 1, the bipartite network regroups all genotypes and phenotypes. Interestingly, we note that our networks are generally dense, with high  $\bar{k}$  in the projections. In addition, we show that there is no clustering structure to speak of in the genotype network, and that the phenotype network is highly clustered. Finally, the phenotype network has a short APL. The high CC and the short APL of the phenotype network hints at interesting community properties in the phenotypes, which clusters phenotypes that are densely connected in subsets that share common biological or genetic information. This finding is in line with a previous, much more applied study on the Human Phenotype Network (HPN). In Darabos et al. (2013), we report that the HPN does have a strong community structure, and we are now reassured to note that the CA model phenotype network shares this characteristic.

### Robustness, Evolvability, and Accessibility

In this section, we analyze the complex relationships between robustness and evolvability, both genotypic and phe-

notypic. We also report results of the influence of phenotypic accessibility. These links are, we believe, the most biologically relevant and could confer the most relevance to our model. In order to conduct this analysis, we study the statistical characteristics of the genotype and phenotype spaces, assigning robustness, evolvability, and accessibility “scores” to each genotype and phenotype, according to the description found in the Methods Section.

Firstly, we report in Figure 4 the strong, quasi linear positive correlation between the number of phenotypes mapping to a genotype, and the evolvability of those genotypes. This correlation is to be intuitively expected from biological organisms in which genotypes responsible for more phenotypes are also considered the most evolvable.

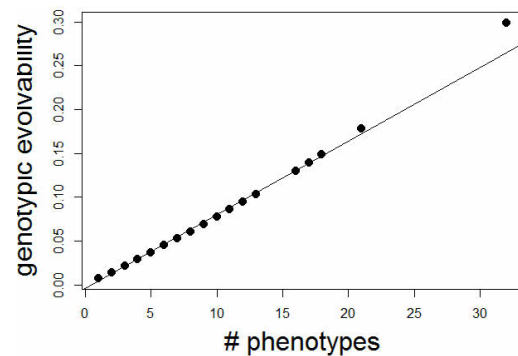


Figure 4: Strong correlation between the number of phenotypes mapped from a genotype and the genotypic evolvability.

One interesting aspect of robustness and evolvability in all systems, is their relationship to the perceived complexity of the phenotype. In our case, the complexity of the phenotype is measured by the size (or length) of the attractor. These relationships are depicted in Figure 5.

In Figure 5, we report a strong negative correlation between the length of the phenotype’s attractor and its robustness. This result agrees with Kauffman’s work on RBNs (Kauffman, 1969), who early on has reported that longer attractors are less stable in RBN systems. The same negative correlation appears with phenotypic evolvability. We speculate that more complex phenotypes have more difficulties accessing other phenotypes because of the basin of the attraction size. This is confirmed by the fact that more robust phenotypes are also less accessible (see below, Figure 7d).

We are especially interested in genetic robustness and evolvability and how they are distributed over the entire genotype space. These results are reported in Figure 6a-b. Moreover, we study the relationship between genetic robustness and evolvability in Figure 6c.

The “bell shaped” distributions of genotypic and phenotypic robustness (not shown here), or neutrality, are also aligned with results in similar studies (Hu et al., 2011). Most

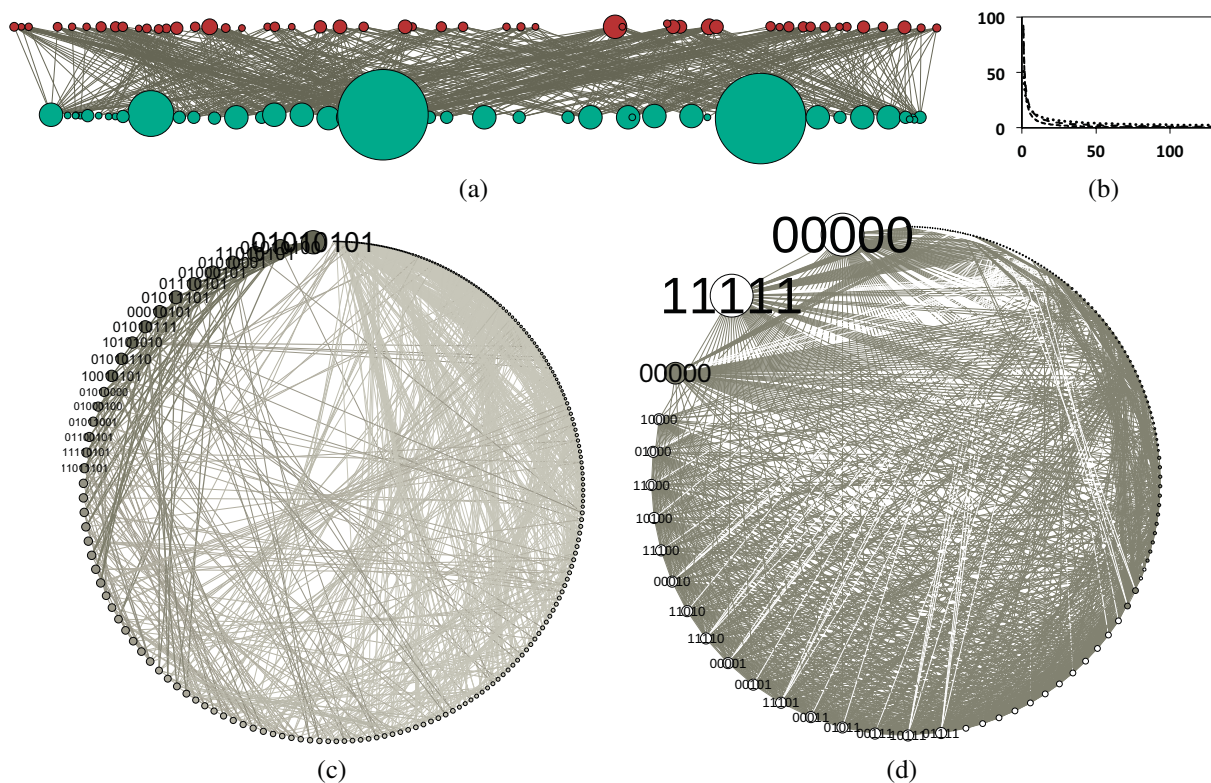


Figure 3: Filtered Bipartite G2P Network, Degree Distributions, and Projections. (a) The top row vertices represent the genotypes, and the bottom row vertices represent the mapped phenotypes. The vertex size is proportional to the degree (i.e. to the number of mapped phenotypes, or mapping genotypes respectively) Vertices of a degree below five are omitted for readability reasons. (b) the “heavy-tailed” degree distribution for the bipartite network and both phenotype and genotype projections. (c) the projection in the genotype space, the vertex sizes are proportional to the number of associated phenotypes, the darker the edge, the more phenotypes the genotypes have in common. (d) the projection in the phenotype space, with vertex sizes proportional to the number of mapping genotypes. The vertex is white if the phenotype is a point attractor, and grey if the attractor is longer (i.e. the phenotype is more complex). The darker the edge, the more genotypes the phenotypes have in common.

genotypes have a high robustness, and a rather low evolvability. However, the genetic evolvability distribution is right-skewed, with a heavy tail, therefore, a small number of genotypes have a very high evolvability. Interestingly, genotypes which have a middle range value robustness ( $\approx 0.5$ ) tend to show a higher degree of evolvability. In other words, high or low genetic robustness yield lower evolvability than mid-range values.

Relating this research to biological organisms, we are particularly interested in the phenotypic aspect of the CA models and how they respond to external (i.e. environmental) and internal (i.e. genotypic) perturbations, and how these perturbations impact robustness. Therefore, we consider the phenotypic robustness of our systems, and study its relationships to genotypic and phenotypic evolvability, genotypic robustness, and the accessibility of the phenotypes. These results are reported in Figure 7.

The first, and natural correlation we are interested in is phenotypic robustness to evolvability. Figure 7a shows a

strong correlation between phenotypic robustness and evolvability. This means that phenotypes that are more robust are also more evolvable, and vice-versa. Although it seems counterintuitive that more stable phenotypes are also more evolvable, our findings are in perfect agreement with Wagner’s conjecture and work, in which stability in biological organisms actually supports evolvability (Wagner, 2008b). In Figure 7b, we see that more robust genotypes do not give rise to more robust phenotypes, and vice-versa. This is not actually surprising, although it is not in line with Hu’s work (Hu et al., 2011). In our work, the robustness of the genotype does not guarantee that of the phenotype. Panel (c) shows that phenotypic robustness and genotypic evolvability are moderately negatively correlated, which indicates that more stable phenotypes are obtained by genotypes that can give rise to fewer phenotypes after a small perturbation. Finally, Figure 7d reveals that phenotypes that are the least accessible from other phenotypes, through small mutation of their genotypes, are also the most stable. This implies that ran-

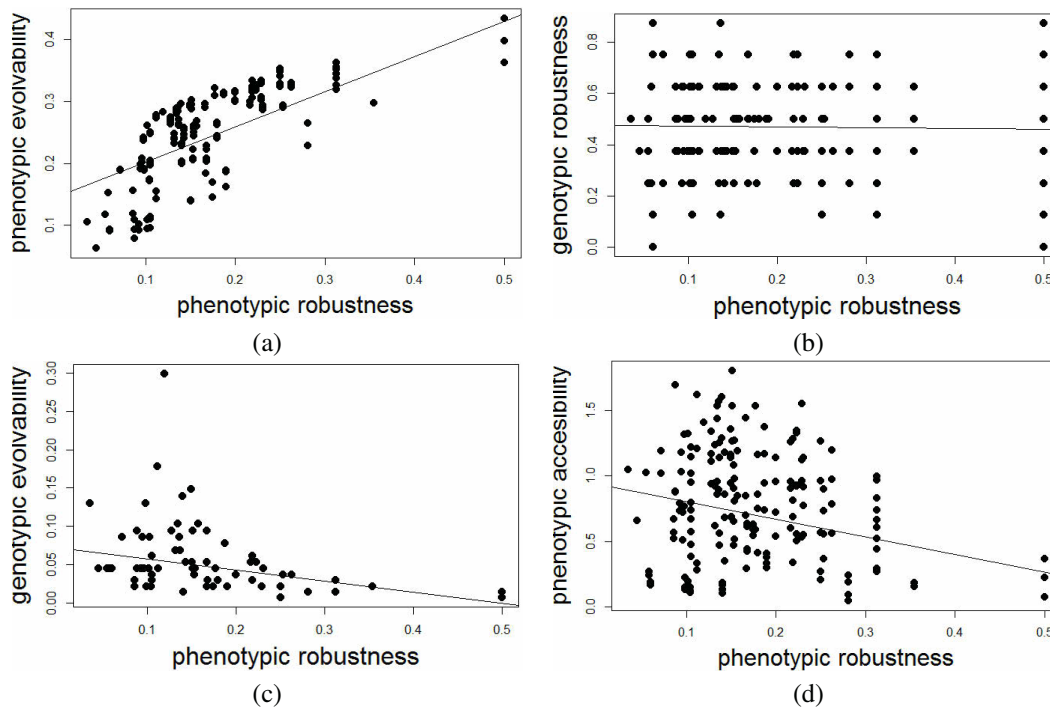


Figure 7: Phenotypic Robustness in relationship to (a) genotypic evolvability ( $r = -0.46$ ), (b) genotypic robustness ( $r = -0.02$ ), (c) phenotypic evolvability ( $r = 0.79$ ), and (d) phenotypic accessibility ( $r = -0.37$ ). We show trend lines on all panels and specify the correlation coefficients ( $r$ ) in parenthesis.

dom mutations are less likely to lead to robust phenotypes than to non-robust phenotypes. Combining the observations in panels (a) and (d), we hypothesize that the most robust phenotypes are difficult to find (Figure 7d) and highly evolvable (Figure 7a).

## Conclusions

In order to explore the complex genotypes-to-phenotypes (G2P) relationship in biological organisms, we propose to use small, simplistic, proxies in the form of a Cellular Automata (CAs) model. CAs share the many-to-many G2P mappings of living organisms. In this implementation, we keep the systems small enough so that we can exhaustively explore both the genotype and the phenotype space, and all initial conditions. To visually assess the G2P associations, we use a bipartite network and its projections onto either space. It shows a dense network, with some prominent genotypes that can access all phenotypes, and some highly popular phenotypes that can be reached by all genotypes.

Genotypes that are associated with the most phenotypes are also the most evolvable, which suggests that evolvable genotypes cluster together and tend to be genetically close. In agreement with Kauffman's work on RBNs, the phenotype's complexity (i.e. attractor length) is negatively correlated with both its evolvability and its robustness. In addition, we conclude that the more evolvable genotypes yield

a robustness in the middle of the range of possible values. These results suggest that some robustness is necessary to promote evolvability in genotypes, but too much robustness will hinder genotypic evolvability.

Finally, we showed that phenotypic robustness and evolvability are closely related, which agrees with Wagner's results, and that phenotypic robustness supports phenotypic evolvability. Despite its counterintuitive nature, this finding is aligned with biological systems. We also see that stable phenotypes are not generated by stable or evolvable genotypes. Moreover, we note that popular (accessible) phenotypes tend to be less robust, which contradicts some evidence in natural systems.

Robustness, evolvability and accessibility correlations, or lack thereof, are highly system-dependent. In the case of CAs, we show that these relationships are, generally, in line with observations in biological organisms. However, the robustness-evolvability-accessibility relationship in biological organisms is also highly system-dependent. Nevertheless, models are unavoidable when studying interactions as complex as G2P, and CAs remain a viable option, considering that they also model the many-to-many G2P relationship.

In a future line of research, we plan on exploring the G2P relationships in much larger CAs in order to inch closer to simple biological organisms. Unfortunately, this will come



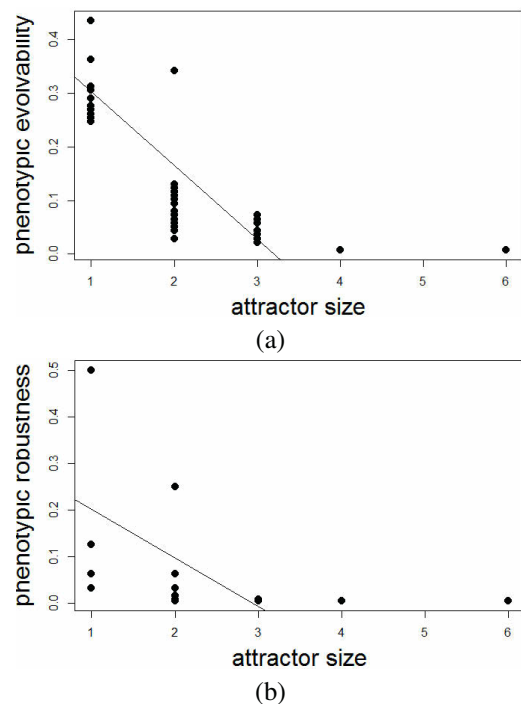


Figure 5: The effect of Attractor size on (a) phenotypic evolvability ( $r = -0.74$ ) and (b) phenotypic robustness ( $r = -0.39$ ). We show trend lines on all panels and specify correlation coefficients ( $r$ ) in parenthesis.

at the cost of the exhaustive search, and force us to select a limited number of sample genotypes and phenotypes mappings. Moreover, we are planning on addressing the phenotypic plasticity, or adaptability of the phenotype during its lifetime, and explore ways to integrate this aspect into our CA model without losing its attractive simplicity. Finally, it would be interesting to compare our results with a real-life biological case, if the quality of the data permits.

### Acknowledgements

This work was partially supported by NIH grants AI59694, LM010098 and LM009012.

### References

- Banzhaf, W., Davidor, Y., p. Schwefel, H., and (eds, R. M. (1994). Genotype-phenotype-mapping and neutral variation - a case study in genetic programming.
- Bloom, J. D., Labthavikul, S. T., Otey, C. R., and Arnold, F. H. (2006). Protein stability promotes evolvability. *Proc Natl Acad Sci U S A*, 103(15):5869–5874.
- Codd, E. F. (1968). *Cellular Automata*. Academic Press, New York.
- Cowperthwaite, M. C., Economo, E. P., Harcombe, W. R., Miller, E. L., and Meyers, L. A. (2008). The ascent of the abundant: how mutational networks constrain evolution. *PLoS Comput Biol*, 4(7):e1000110.

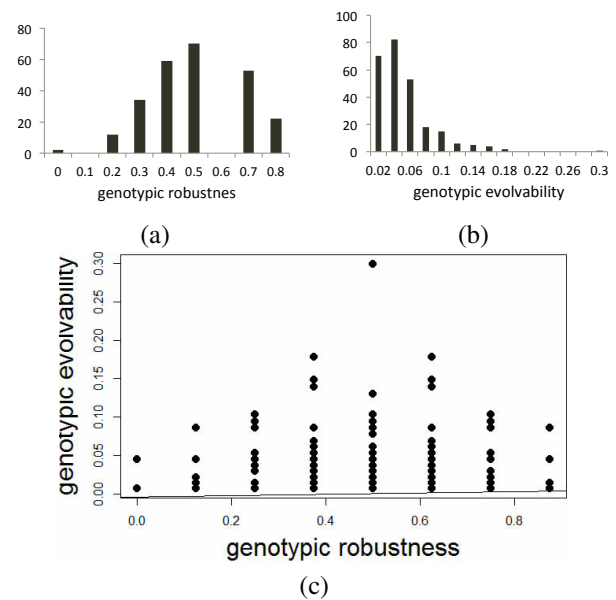


Figure 6: Genetic Robustness and Evolvability and their relationship to one another. Top row: distribution of (a) genetic robustness and (b) genetic evolvability over all possible genotypes. Bottom figure: the relationship, with a slight positive correlation between robustness and evolvability (correlation coefficient  $r = 0.10$ ).

Darabos, C., Desai, K., Cowper-Sallari, R., Giacobini, M., Graham, B., Lupien, M., and Moore, J. (2013). Inferring human phenotype networks from genome-wide genetic associations. In Vanneschi, L., Bush, W., and Giacobini, M., editors, *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, volume 7833 of *Lecture Notes in Computer Science*, pages 23–34. Springer Berlin Heidelberg.

Ferrada, E. and Wagner, A. (2008). Protein robustness promotes evolutionary innovations on large evolutionary time-scales. *Proceedings of the Royal Society London B*, 275:1595–1602.

Hu, T., Payne, J. L., Banzhaf, W., and Moore, J. H. (2011). Robustness, evolvability, and accessibility in linear genetic programming. In *Proceedings of the 14th European conference on Genetic programming*, EuroGP’11, pages 13–24, Berlin, Heidelberg. Springer-Verlag.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22:437–467.

Kauffman, S. A. and Weinberger, E. D. (1989). The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of Theoretical Biology*, 141:211–245.

Newman, M. (2010). *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA.

Nuzhdin, S. V., Friesen, M. L., and McIntyre, L. M. (2012). Genotype-phenotype mapping in a post-gwas world. *Trends Genet*, 28(9):421–426.



- Ochoa, G., Tomassini, M., Vérel, S., and Darabos, C. (2008). A study of nk landscapes' basins and local optima networks. *CoRR*, abs/0810.3484.
- Wagner, A. (2005). *Robustness and Evolvability in Living Systems*. Princeton University Press.
- Wagner, A. (2008a). Neutralism and selectionism: a network-based reconciliation. *Nat Rev Genet*, 9(12):965–974.
- Wagner, A. (2008b). Robustness and evolvability: a paradox resolved. *Proc Biol Sci*, 275(1630):91–100.

# The Effects of Assortment on Population Structuring Traits on the Evolution of Cooperation

Adam Jackson<sup>1</sup> and Richard Watson<sup>1</sup>

<sup>1</sup>The University of Southampton, Southampton SO17 1BJ, UK  
aj3e10@soton.ac.uk

## Abstract

Population structure plays an important role in the evolution of social behaviours, particularly by generating positive assortment on social interactions. This enables cooperative behaviours that have a net cost to the individual to spread by directing their benefits towards other cooperators. Previous work on the coevolution of population structures and social behaviours has suggested that the evolution of population structuring traits is strongly influenced by the dominant social strategies. Here we investigate the idea that the coevolution of population structure and behaviour can be enhanced in favour of cooperation when there is also assortment on the population structuring traits themselves. This paper presents a simulation model that investigates the effects of evolving this second-order assortment and introduces a mathematical framework to model it in terms of the replicator equation. We find that with second order assortment the dominant social behaviour trait does not necessarily have to control the evolution of population structure, increasing the range of social scenarios in which population structures that support increased cooperation can evolve.

## Introduction

Population structures that promote positive assortment on social behaviours are a key pathway to the evolution of cooperation (Nowak and May, 1992). When cooperators interact disproportionately with other cooperators then the benefits of the cooperative acts will fall predominantly on cooperators, raising their fitness while reducing the risk of defectors taking the benefits without assuming the costs. There are many mechanisms that can support the evolution of cooperation, including genetic relatedness, signalling (greenbeard effects) and reciprocity through repeated interactions. The common principle at work is that these mechanisms generate correlated interactions by co-operators (Godfrey-Smith, 2009). Only when the benefits of cooperation are directed at other cooperators in this way can cooperative behaviours that benefit the recipient at a cost to the individual spread (Hamilton, 1964; Lehmann and Keller, 2006).

This is important both to explain the otherwise surprising ubiquity of cooperative behaviours and because of the underlying importance of altruism to the major transitions in

evolution, the processes through which new levels of biological organisation emerge as previously independently replicating entities become higher-order individuals that must reproduce as a unit (Maynard Smith and Szathmáry, 1997). The altruistic sacrifice of individual reproductive capability at the lower level is such a characteristic of this process that the major transitions can be viewed as extreme examples of social integration (Michod, 2000; Queller and Strassmann, 2009; Bourke, 2011).

Traditionally social evolution models have investigated the evolution of cooperation against a fixed population structure. But many organisms possess individual-level genetic traits that affect population structure, such as a group size preferences or dispersal radius (Pepper and Smuts, 2002). Recent work has begun to look at how population structuring traits and social behaviours can coevolve in a process of *social niche construction* (Powers et al., 2011). Artificial Life models of the coevolution of group size preferences and social behaviours have shown that linkage disequilibrium develops connecting a preference for small groups with cooperative social traits (Powers and Watson, 2011). In such circumstances between group effects can outweigh within group competition due to the high variance between groups, caused by sampling small groups from a large population (Wilson and Colwell, 1981), leading to a positive feedback loop in favour of increasing levels of cooperation and decreasing group size even when the social game is unfavourable to co-operators.

It is generally well accepted then that positive assortment can enable cooperation to prevail when it would not in a well-mixed population. In previous work we have been investigating the factors that lead population-structuring traits to support the evolution of cooperation using evolutionary game theory. Many changes in population structure can be represented as transformations to the social game being played, including reciprocity, kin selection (Taylor and Nowak, 2007) and group structure (Van Veelen, 2011). We have looked at the coevolution of population structures and social behaviours through abstract models of *metagames* in which populations evolve not just their social behaviours but

also the payoff matrix of the game they are playing. These metagame models have demonstrated that the strongest influence on the selective pressure on the population structure is the dominant social strategy; when co-operators are the dominant social type population structures more favourable to cooperation evolve, and likewise when defectors dominate. This is in apparent contradiction with the logical arguments that support social niche construction (Powers et al., 2011), and would imply a limited causal significance for population-structuring traits if they only increase the spread of cooperative behaviours in situations where co-operators are already favoured.

Here we suggest that one of the keys to the strength of social niche construction is that not only do population-structuring traits (*PSTs*) induce assortment on the social behavioural traits but that there is also a degree of assortment on the *PSTs* — a *second order assortment*. In the example of a group size preference trait, if the group size preference leads to a greater chance of living in a group of the desired size it also implicitly creates groups composed of individuals with similar group size preferences (Powers et al., 2011). Not all population-structuring traits do result in second-order assortment. It is argued that the key distinction between kin-selection and greenbeard traits is that relatedness leads to the same average measure of correlation for all the genes of an organism while greenbeard traits only lead to correlation on a few genes (Ridley and Grafen, 1981; Bourke, 2011). As a consequence greenbeard traits may be susceptible to parasitism (Okasha, 2002) and intragenomic conflict — suppressor mutations arising at other *loci* (though others have argued that this is not inevitable as selection for greenbeard and greenbeard-imitation traits may be aligned (Gardner and West, 2010; Biernaskie et al., 2011)). So we might expect that when assortment is generated by relatedness there would be the same degree of assortment on population-structuring and behavioural traits, while if it was generated by a greenbeard signal this is not necessarily the case.

In this paper then we investigate the effects of second-order assortment on the evolution of population-structuring traits that support cooperation, in particular the nature of the second-order assortment. We do this through an artificial life model in which different levels of assortment are expressed literally — assorters are physically more likely to interact with each other. We also develop a novel way to model these results mathematically by modifying the replicator equation. We show that when second-order assortment is random or absent then the spread of population-structuring traits that support increased cooperation is strongly linked to the success of co-operators. When population-structuring traits affect both assortment on social behaviours and on themselves, then the conditions under which populations can evolve population structures beneficial to cooperation are enlarged.

## Social Dilemmas in Evolutionary Game Theory

The standard mathematical tool for analysing the evolution of social behaviours is evolutionary game theory (Maynard Smith, 1982). Evolutionary game theoretic models are appropriate when the fitness of a strategy depends on the frequency with which it and other strategies are found in the population as well as the inherent qualities of the strategy — as is the case for social behaviours. There are a number of ways to interpret game theoretic models; here we think of a population divided into a  $n$  genetically determined types corresponding to pure strategies of the game. The fitness payoffs for the interactions between types determines the payoff matrix of the game. The changing frequencies of the different strategies are modelled using the *replicator equation* (Taylor and Jonker, 1978).

$$\dot{x}_i = x_i(f_i(\mathbf{x}) - \bar{f}(\mathbf{x})) \quad (1)$$

Where  $x_i$  is the frequency of the  $i$ -th strategy,  $f_i$  is the fitness of that strategy given a population state vector  $\mathbf{x} = (x_1, \dots, x_n)$  and  $\bar{f} = \sum_{i=1}^n x_i f_i(x)$  is the mean fitness of the population given that state. The stable equilibria of the population state under the replicator equation determine *evolutionarily stable states* (ESS) to which the population will return if the frequencies are subject to small perturbations.

Here we focus on social interactions with two social strategy types — co-operators and defectors. Although they are the simplest types of game, these two strategy games include the canonical social dilemmas such as the Prisoner's Dilemma. An arbitrary such game can be defined by the  $2 \times 2$  payoff matrix  $G = \begin{pmatrix} R & S \\ T & P \end{pmatrix}$ .  $R$  is the reward for mutual cooperation,  $P$  the punishment for mutual defection,  $T$  the temptation to defect and  $S$  the suckers payoff. The two strategies are labelled  $C$  for co-operate and  $D$  for defect. We impose the condition that  $R > P$  as the benefits of mutual cooperation are assumed to outweigh those from mutual defection. The complete four-dimensional space of games is determined  $R, S, T$  and  $P$ ; varying their relative magnitudes leads to a diverse range of social scenarios. By normalising  $R$  to 1 and  $P$  to 0 it is possible to project this space onto to a two dimensional plane parameterised by  $S$  and  $T$  (Santos et al., 2006) that we call the  $ST$ -plane. This projection is done via the transformation:

$$\begin{pmatrix} R & S \\ T & P \end{pmatrix} \rightarrow \begin{pmatrix} \frac{R-P}{R-P} & \frac{S-P}{R-P} \\ \frac{T-P}{R-P} & \frac{P-P}{R-P} \end{pmatrix} = \begin{pmatrix} 1 & S' \\ T' & 0 \end{pmatrix} \quad (2)$$

This transformation has the important property that every game with  $R > P$  is equivalent to a game on the  $ST$ -plane — there is a corresponding game that has the same dynamics (though the speed of selection may change). This is because the transformation multiplies the replicator equation

by a constant, leaving the evolutionary stable states determined by the equation's fixed points unchanged. The projection is valid whenever  $R \neq P$  (which is always true for the games we are interested in as  $R > P$ ). This makes the  $ST$ -plane an extremely useful tool to aid conceptual understanding as a representative subset of the two-strategy social dilemmas. The convention is to assume that mutual cooperation is preferable to unilateral cooperation ( $R = 1 > S$ ) and to an equal probability of unilateral cooperation or defection ( $2R = 2 > T + S$ ) (Macy and Flache, 2002), hence the  $ST$ -plane is plotted for the region  $-1 \leq S \leq 1$ ,  $0 \leq T \leq 2$ . Note that we assign a neutral payoff  $P = 0$  to mutual defection; recent work has suggested that ecological constraints leading to negative payoffs for one or both of  $R$  and  $P$  may lead to alternate pathways to altruism in the absence of population structure (Doncaster et al., 2013).

Social dilemmas essentially occur when there is a conflict between the rational outcome and the Pareto-efficient outcome, the individually rational choices for each player leading to a deficient outcome for both. This can arise due to *greed*, the difference between unilateral defection and mutual cooperation ( $T - R$ ), or *fear*, the difference between unilateral cooperation and mutual defection ( $S - P$ ), or both (Santos et al., 2006). These two factors correspond to the two axes of the  $ST$ -plane. The lines  $S - P = 0$  and  $T - R = 0$  split the  $ST$ -plane into four quadrants corresponding to four fundamental two player games that cover the most common types of conflict (Figure 1):

The *Harmony game* ( $R > T > S > P$ ), the least seen of the four as there is no social dilemma — group and individual interests are aligned with cooperation always the most successful strategy. A population of all cooperators is the ESS.

The *Prisoner's Dilemma* ( $T > R > P > S$ ) where the ESS is a population of no cooperators. The Prisoner's Dilemma is a particularly important example of a game and it models scenarios in which individual selection results in defection, to the detriment of the population as a whole.

The *Snowdrift game* ( $T > R > S > P$ ), an anti-coordination game in which interactions with other strategies carry higher payoffs than same-strategy interactions. The Snowdrift game is thus significant as the only game that sustains a stable polymorphic population — the evolutionarily stable state has  $\frac{S-P}{S+T-R-P}$  cooperators.

The *Stag-Hunt game* ( $R > T > P > S$ ), a coordination game in which populations of all cooperators and no cooperators are both ESS. There is an unstable equilibrium with  $\frac{S-P}{S+T-R-P}$  of the population cooperators which divides the two basins of attraction; this is the only game in which the initial frequency of cooperators is significant in determining the ESS.

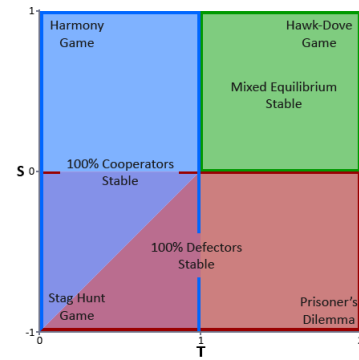


Figure 1: The  $ST$ -plane showing the four fundamental games and how they are determined by the regions in which the different ESS exist.

## Model Details

To investigate the effects of population-structuring trait assortment on the evolution of population structures favourable to cooperation we created an evolutionary algorithm based on a fixed size population of asexual agents living in non-overlapping generations. A range of potential assorting behaviours are abstractly represented by having agents physically cluster to varying extents determined by their population-structuring traits. This physical assortment influences fitness by affecting who each agent plays with, specifically as each agent's fitness is determined by playing a specified evolutionary game  $G$  against its nearest neighbour.

The model consists of a population of  $P = 100$  agents living in a continuous space that is topologically toroidal. Each agent is represented as a circle  $r = 4$  units in radius. The dimensions of the world is  $100r \times 100r$  units. Each agent's genotype is haploid with two *loci* — a social trait gene that determines its social behaviour (with either 'co-operator'  $C$  or 'defector'  $D$  alleles) and a population-structuring trait with alleles  $A$  for assorting behaviours and  $M$  for freely mixing behaviour. This gives four genotypes:  $CA$ ,  $DA$ ,  $CM$  and  $DM$ . The model is initially populated with these four genotypes present at equal frequency with agents placed randomly in the world.

There are two stages to the evolutionary algorithm. First the agents selectively aggregate for  $T = 10000$  timesteps. Then each member of the population plays a game with its nearest neighbour to determine its fitness. 'Nearest neighbour' is not a symmetric relationship, so while all agents play at least one game (against their own nearest neighbour), some may be played against multiple other times by other agents that have it as their nearest neighbour. As all the games here are symmetric this is equivalent to the agent playing multiple times with different opponents. For this model, an agent's fitness was defined to be its average payoff received per interaction rather than cumulative payoff;



we chose not to reward or penalise an agent for being involved in multiple games. Recording the payoff of the focal player would have also achieved this result but would mean a co-operator closest to another co-operator but surrounded by defectors received the same payoff as a co-operator with no defectors near it; however, there are arguments in favour of other mechanisms and the choice of average payoff represented a trade-off. After the fitnesses were calculated the population was reproduced using tournament selection up to the fixed size  $P$  again: for  $P$  repetitions two agents were drawn from the population and the agent with the highest fitness (or a random agent if they had equal fitness) is reproduced clonally with a small chance of mutation. To represent the intuition that population structure evolves more slowly than social behaviours, the mutation rate for the social behaviour allele was set to a probability  $m_{SB} = 0.01$ , while the probability of a mutation of the population-structuring trait was  $m_{PST} = m_{SB}/2$ .

The agents' movement is modelled by a variation on gravitational attraction. To simplify calculations, each agent is defined to have a mass of 1. The 'gravitational' force between agents  $i$  and  $j$  is then calculated as  $\frac{G_{i,j}}{d^2}$ . Here  $d$  is the distance between the two agents' centers taking into account the fact that the world is toroidal.  $G_{i,j}$  is the attractive constant between the two agents, which is determined by their genotypes and three parameters -  $\alpha$ ,  $\beta$  and  $\gamma$ . These parameters influence the levels of first and second-order assortment in the model:

- $\alpha$  is the attractive force that agents with the assorting allele  $A$  feel towards agents with the same social trait as themselves. So agents with the genotype  $CA$  feel an attractive force of strength  $\alpha$  towards other  $CA$  and  $CM$ -typed agents.
- $\beta$  is the attractive force between agents with the assorting PST allele  $A$ .
- $\gamma$  is the attractive force between agents with the mixing PST allele  $M$ .

Figure 2 shows diagrammatically the different forces that exist between each genotype. The combined forces can be tabulated to give the strength of attraction between genotypes (Table 1). Note that the attractive forces are not symmetrical, unlike real models of gravitation — agent  $i$  may be attracted to agent  $j$  more than  $j$  is to  $i$ .  $G_{i,j}$  is then computed as  $rg$  where  $g$  is the attractive force from agent  $i$  to agent  $j$  as given in the table.

There is also a repulsive force with magnitude -1 that effects the agents when they are closer than  $2r$  apart that prevents the agents from overlapping. At each timestep the forces between all agents are calculated and the net force  $F_i$  for each agent calculated. Friction is then accounted for using the equation  $F_i = F_i - 0.2v_i$  where  $v_i$  is the agent's

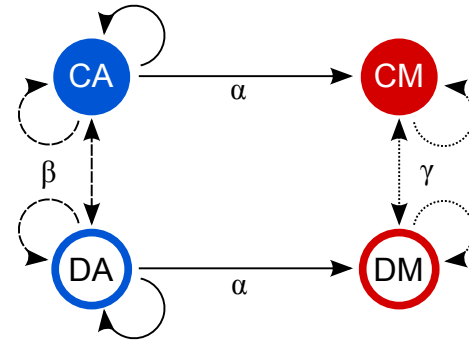


Figure 2: The attraction between the different genotypes is determined by the three parameters  $\alpha$ ,  $\beta$  and  $\gamma$

	CA	DA	CM	DM
CA	$\alpha + \beta$	$\beta$	$\alpha$	0
DA	$\beta$	$\alpha$	0	$\alpha$
CM	0	0	$\gamma$	$\gamma$
DM	0	0	$\gamma$	$\gamma$

Table 1: The attractive force from an agent of one genotype (rows) to an agent of another genotype (columns).

previous velocity. Each agent's acceleration, velocity and position are then calculated and updated using numerical integration and the standard equations of motion in a plane using an integration timestep of 0.1.

## Model Scenarios

The model was run using four different sets of parameters, the relative strengths of  $\alpha$ ,  $\beta$  and  $\gamma$  defining four different scenarios with respect to the levels of first and second-order assortment and the nature of the second-order assortment:

**No Assortment** ( $\alpha = 0$ ,  $\beta = 0$ ,  $\gamma = 0$ ). In this control scenario there is no attraction between agents. Consequently each agent's nearest neighbour is randomly determined by the initial placement of agents in the world and frequencies of the social strategies are expected to reach the equilibrium of the game being played.

**First-Order Assortment Only** ( $\alpha = 1$ ,  $\beta = 0$ ,  $\gamma = 0$ ). In this scenario, the only assortment is between social behaviours, where agents with the assorting population-structuring trait are attracted to agents with the same social strategy traits, regardless of whether or not the other agent possesses the assorting trait.

**Emergent Second-Order Assortment** ( $\alpha = 1$ ,  $\beta = 1$ ,  $\gamma = 0$ ). In this model agents with the assorting trait are attracted to others with the same strategy, but there is also attraction between agents with the assorting trait and other agents with the assorting trait. This produces a scenario in

which second-order assortment is tied to the mechanism that generates first-order assortment, as may be the case where strategy-assorters possess greenbeard traits.

**Enforced Second-Order Assortment** ( $\alpha = 1, \beta = 1, \gamma = 1$ ). In this model as well as first order assortment there is also uniform assortment on population-structuring traits. This is intended to produce a situation in which levels of assortment are uniform across different traits such as may be the case with relatedness.

## Results

Clustering occurs in the model as the agents aggregate. In the control scenario, no clustering takes place. In the scenario with only first-order assortment, cooperators cluster with cooperators and defectors with defectors. Predominantly though it is agents with the *CA* genotype grouping with other *CAs* and *DAs* grouping with *DAs* as these are the agents attracted to others with the same strategy allele. In the second variation cooperators and defectors cluster together, with *CAs* and *DAs* at the heart of the clusters and *CMs* and *DMs* at the edges attracted to other cooperators and defectors respectively. In the third scenario where second order assortment is enforced the clusters are more mixed.

To investigate the effects of second-order assortment over a wide range of social dilemmas we took a  $21 \times 21$  lattice of points on the *ST*-plane spaced 0.1 units apart and ran each scenario for every game on the lattice. Each scenario was repeated 5 times for every game and the results averaged. One of the dynamics in the model is that there is no selective difference between individuals with the same social strategy allele in the absence of individuals with the other social strategy allele. The difference in fitness between, for instance, the *CA* and *CM* genotypes comes from their different interactions with the *DA* and *DM* genotypes. So when the model reaches a state in which only one of the social strategy alleles is present then there is no selective pressure between them and their frequencies begin to take a random walk. Experimental testing indicated 20 generations proved a balance between letting the model reach equilibrium and mitigating the effects of the random walk, so this was the length of each run of the model.

Our hypothesis was that increasing second-order assortment would increase the spread of the assorting population-structuring trait. We found some support for this view but with complications, some of which were obvious in retrospect. Figure 3 plots the mean absolute frequency of the *C* allele over the *ST*-plane in all four models. In all scenarios in which there is assortment, cooperators perform better than in the control. However, counter to our initial expectations, cooperation is more successful when there is just assortment on social strategies.

Figure 4 plots the mean frequencies of the *A* allele over

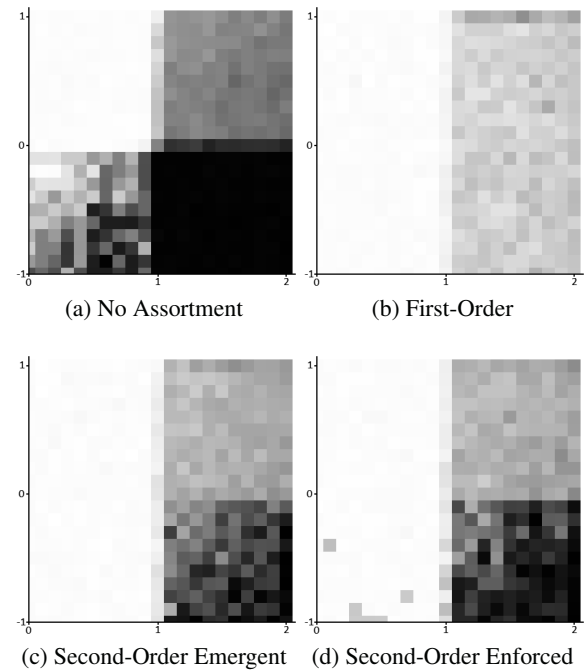


Figure 3: The mean absolute frequency of the *C* allele over the *ST*-plane in all four models on a scale where white indicates 100% cooperators, black 100% defectors.

the *ST*-plane. In the control model there frequency of the *A* is essentially random. In the other models, as expected, increasing levels of assortment increases the spread of the *A* allele. These results are summarised in Table 2 which records the mean frequencies of the *C* and *A* alleles over all games and all runs. As the table shows, there is essentially no net change in frequencies over the whole of the *ST*-plane in the control model. In the three scenarios with assortment, the frequencies of the *C* and *A* alleles increase, but with a trade-off between increased levels of cooperation and assortment on the population-structuring trait.

Scenario	Mn <i>C</i>	Var <i>C</i>	Mn <i>A</i>	Var <i>A</i>
No Assortment	0.505	0.140	0.501	0.005
First-Order	0.814	0.019	0.669	0.017
2 <sup>nd</sup> -order Emergent	0.748	0.088	0.684	0.025
2 <sup>nd</sup> -order Enforced	0.729	0.103	0.731	0.028

Table 2: The mean final frequencies of the *C* and *A* alleles over the *ST*-plane in each scenario.

## Mathematical Model of Altered Interaction Frequencies using the Replicator Equation

Agent-based simulation models like the one presented here are subject to noise and it can be difficult to tune the desired

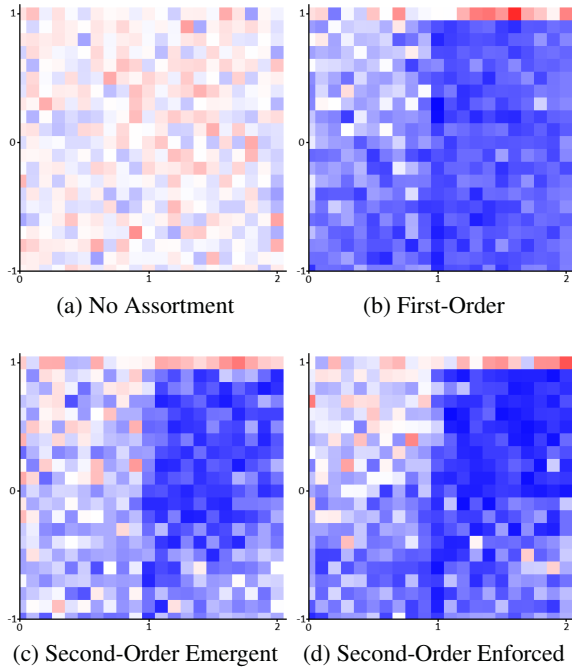


Figure 4: The mean frequencies of the  $A$  allele over the  $ST$ -plane on a red-white-blue scale. Red indicates the  $A$  allele decreases in frequency ( $< 0.5$ ), blue that the  $A$  allele increases in frequency ( $> 0.5$ )

behaviours precisely, so there are benefits to reproducing the results in a mathematical model. Here we present a formalism to do so by modifying the replicator equation. The replicator equation models the evolutionary success of a group of genotypes based on the difference between the fitness of the genotype and the mean fitness of the population. The fitness of a genotype is given by its expected payoff based on social dilemma it is engaged in — the payoff for its interaction with each other genotype multiplied by the probability of interacting with that genotype. When the population is freely mixed, these interaction probabilities match the frequencies at which the genotypes occur in the population, but they can be changed by population-structuring mechanisms.

A population state vector  $x \in S_n$  is a vector of the frequencies of different genotypes in a population; alternatively it can be viewed as the probabilities of interacting with a given genotype. The entries of the vector must sum to 1 so it is defined on the simplex  $S_n = \{(x_1, \dots, x_n : \sum_i^n x_i = 1)\}$ . We can define a family of  $n$  *interaction functions*  $e_i : S_n \rightarrow S_n$  that map the population vector to the actual frequencies at which the  $i$ -th genotype encounters other genotypes. The fitness of the type is the composition of the fitness and interaction functions  $f_i \circ e_i(x)$ , and the mean fitness is  $\sum_j^n x_j f_j \circ e_j(x)$ . This gives the new replicator equation:

$$\dot{x}_i = x_i \left( f_i \circ e_i(x) - \sum_j^n x_j f_j \circ e_j(x) \right) \quad (3)$$

The key idea here is that essentially whenever the replicator equation is used there an implicit interaction function, but in a well-mixed population the interaction functions are just the identity. This generalises the replicator equation by explicitly recognising other interaction functions. The set of fitness functions  $g_i : g_i = f_i \circ e_i$  then defines a new game that when played in a well-mixed population is equivalent to the original game played with the given population structure. If the interaction functions can be represented by matrices then they can be used to find the payoff matrix for the transformed game, but that does not apply in this instance where the interaction functions are non-linear.

The non-linearity of the interaction functions in practise is often forced by the requirement that they map valid population states to valid population states (where the entries sum to 1). In general any map of  $S_n$  can be normalised to one of  $S_n$  to itself by dividing the resulting vector by the sum of its entries, but this results in non-linearity. As long as the interaction functions are continuous though the replicator equation can still be used.

### Modelling the Results with Interaction Functions

We used the formalism of interaction functions to mathematically model the results of the simulated scenarios. This was done by constructing interaction functions based on data from the simulations. First we define the four strategy game in the absence of interaction functions. If we consider the strategies to be  $x_1 = CA$ ,  $x_2 = DA$ ,  $x_3 = CM$ ,  $x_4 = DM$  then for an arbitrary social game  $G = \begin{pmatrix} R & S \\ T & P \end{pmatrix}$  the matrix of the full four strategy game is:

$$\begin{pmatrix} R & S & R & S \\ T & P & T & P \\ R & S & R & S \\ T & P & T & P \end{pmatrix} \quad (4)$$

This is then modified using interaction functions. To represent the changed number of interactions due to the population structure we use a simple non-linear interaction function — multiplying each entry in the population state vector by a scalar representing an increased chance of encountering that genotype and then normalising the resulting vector so the entries sum to 1. The scalars were calculated by running the model until the first reproduction event (at  $T = 10000$ ) 100 times starting from evenly distributed population frequencies and recording the total number of games played between each pair of genotypes. Dividing this by the total number of interactions gave the mean frequencies at which a given genotype would encounter each other genotype when the actual frequency of each genotype was 0.25, so dividing

again by 0.25 gives the the actual encounter rate between different genotypes as a multiple of what would have been expected in a well-mixed population. We used these scalars to define the interaction functions for the four types in the model. This was a basic way of determining the interaction functions — a more complex way would have been to generate scalars for different actual population frequencies and interpolate between them to create more complex interaction functions. However, this simple method was sufficient to capture the behaviour of the three non-control models; the match between the results is illustrated in Figure 5.

	CA	DA	CM	DM
CA	2.32	0.14	1.19	0.36
DA	0.13	2.35	0.33	1.18
CM	1.54	0.44	0.93	1.09
DM	0.46	1.53	1.09	0.92
CA	2.14	0.93	0.72	0.20
DA	0.93	2.15	0.21	0.70
CM	0.96	0.28	1.22	1.54
DM	0.26	0.94	1.55	1.25
CA	2.22	1.07	0.49	0.22
DA	1.06	2.21	0.22	0.50
CM	0.51	0.23	1.50	1.76
DM	0.24	0.54	1.77	1.45

Table 3: The rows define the interaction functions giving the coefficients that modifying how likely it is for the row genotype to encounter the column genotype for the three scenarios with assortment, listed in order.

Table 3 gives the interaction coefficients that were used to define the interaction functions for the three scenarios in which there was assortment in the model. For example the first row defines the interaction function  $e_1$ , describing the transformation in the interaction frequencies for the genotype  $CA$  in the social strategy assortment-only model:

$$e_1 \begin{pmatrix} x_{CA} \\ x_{DA} \\ x_{CM} \\ x_{DM} \end{pmatrix} = \frac{1}{2.32x_{CA} + 0.14x_{DA} + 1.19x_{CM} + 0.36x_{DM}} \begin{pmatrix} 2.32x_{CA} \\ 0.14x_{DA} \\ 1.19x_{CM} \\ 0.36x_{DM} \end{pmatrix}$$

## Discussion

Our expectations for this model were that increasing levels of assortment on population-structuring traits (PSTs) would lead to the increased prevalence of the PST that supported cooperative behaviours — in this model represented by a PST that directly increased correlated interactions between individuals with the same social strategy. This was true, but we did not anticipate that there would be a degree of trade-off between increased levels of the cooperative ( $C$ ) and assorting ( $A$ ) alleles. The comparison of the different scenarios reveals that cooperative strategies are more successful

when there is just first order assortment, though the frequency of cooperators still increases when there is second-order assortment relative to when there is no assortment at all.

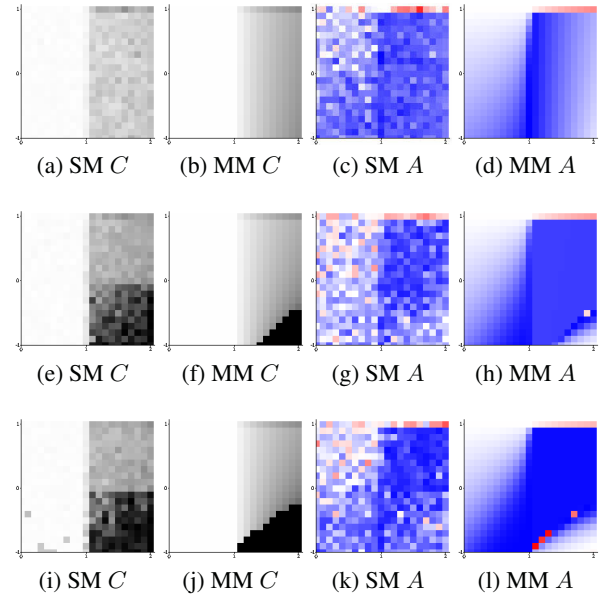


Figure 5: Visual comparison of simulation model (SM) results and the mathematical model (MM) using interaction functions for the three scenarios with assortment. The simulation model results are as in Figures 3 and 4, the mathematical model graphs reproduce these scenarios.

The reason for this trade off is that assortment on social strategy and population structuring traits are not orthogonal processes. When there is just social strategy assortment, cooperators interact preferentially with cooperators and defectors with defectors, greatly reducing the number of cross-strategy interactions. The inclusion of assortment on the population structure alleles reduces this effect by bringing together cooperators and defectors with the same PST allele. While increasing second order assortment decreases the frequencies of co-operators (relative to strategy assortment only), it increases the spread of the PSTs that ultimately promote cooperation. Hence essentially second order assortment decreases the dependency between cooperation and cooperation-promoting PSTs; when there is second-order assortment the  $A$  allele is able to spread even when the local social game is a prisoner's dilemma dominated by defectors.

An alternate way of looking at this is that it demonstrates that the dominant social behavioural trait does not necessarily have to control the evolution of population structure. This is an important result for social niche construction – population structures that support enhanced cooperation must be able to evolve even in conditions unfavourable to coop-



erative behaviours or social niche construction would be a mechanism that just accelerates the evolution of cooperation rather than enabling it where otherwise defection would be favoured. If we imagine that there is a separation of timescales where population structures evolve more slowly than social behaviours then when there is second-order assortment on PSTs, population structures can evolve to become more supportive to cooperation even when the current social dilemmas are dominated by defectors. This would then establish a basis in population structure for cooperative traits to then spread more easily when conditions change to become more favourable.

The model could be extended in a number of ways, such as by allowing for repeated interactions and hence iterated strategies, or examining a wider range of model parameters. The successful realisation of the simulation results in a mathematical model using interaction functions to change the replicator equation also opens up avenues for future work. In particular it is possible to more precisely model different levels of first and second-order assortment using interaction functions; because the assortment in the simulation model is generated by the gravitational attraction it is difficult to tune and potentially presents an issue in comparing the results across different scenarios. Interaction functions have many potential applications that can be pursued beyond this model; they can provide a general mechanism to determine the effective game played when types interact within a population at non-random frequencies and hence allow comparisons between the actual and effective games that are being played by the population in a principled manner. This work also demonstrates that they can be applied to empirical or simulation-derived data to model the results mathematically.

In conclusion, here we have shown that although second-order assortment on population structuring traits can partially disrupt assortment on social behaviours, it increases the range of behaviours in which population structures that support increased cooperation can evolve.

### Acknowledgements

This work was supported by an EPSRC Doctoral Training Centre grant (EP/G03690X/1).

### References

- Biernaskie, J. M., West, S. A., and Gardner, A. (2011). Are greenbeards intragenomic outlaws? *Evolution*, 65(10):2729–2742.
- Bourke, A. (2011). *Principles of social evolution*. Oxford University Press, USA.
- Doncaster, C. P., Jackson, A., and Watson, R. A. (2013). Manipulated into giving: when parasitism drives apparent or incidental altruism. *Proceedings of the Royal Society B: Biological Sciences*, 280(1758).
- Gardner, A. and West, S. A. (2010). Greenbeards. *Evolution*, 64(1):25–38.
- Godfrey-Smith, P. (2009). *Darwinian populations and natural selection*. Oxford University Press, USA.
- Hamilton, W. (1964). The genetical evolution of social behaviour. ii. *Journal of theoretical biology*, 7(1):17–52.
- Lehmann, L. and Keller, L. (2006). The evolution of cooperation and altruism—a general framework and a classification of models. *Journal of evolutionary biology*, 19(5):1365–1376.
- Macy, M. and Flache, A. (2002). Learning dynamics in social dilemmas. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 3):7229.
- Maynard Smith, J. (1982). *Evolution and the Theory of Games*. Cambridge Univ Pr.
- Maynard Smith, J. and Szathmáry, E. (1997). *The major transitions in evolution*. Oxford University Press, USA.
- Michod, R. (2000). *Darwinian dynamics: evolutionary transitions in fitness and individuality*. Princeton University Press.
- Nowak, M. and May, R. (1992). Evolutionary games and spatial chaos. *Nature*, 359(6398):826–829.
- Okasha, S. (2002). Genetic relatedness and the evolution of altruism. *Philosophy of Science*, 69(1):138–149.
- Pepper, J. and Smuts, B. (2002). A mechanism for the evolution of altruism among nonkin: positive assortment through environmental feedback. *The American Naturalist*, 160(2):205–213.
- Powers, S., Penn, A., and Watson, R. (2011). The concurrent evolution of cooperation and the population structures that support it. *Evolution*, 65(6):1527–1543.
- Powers, S. and Watson, R. (2011). Evolution of individual group size preference can increase group-level selection and cooperation. *Advances in Artificial Life. Darwin Meets von Neumann*, pages 53–60.
- Queller, D. and Strassmann, J. (2009). Beyond society: the evolution of organismality. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1533):3143–3155.
- Ridley, M. and Grafen, A. (1981). Are green beard genes outlaws? *Animal Behaviour*, 29.
- Santos, F., Pacheco, J., and Lenaerts, T. (2006). Evolutionary dynamics of social dilemmas in structured heterogeneous populations. *Proceedings of the National Academy of Sciences of the United States of America*, 103(9):3490.
- Taylor, C. and Nowak, M. A. (2007). Transforming the dilemma. *Evolution*, 61(10):2281–2292.
- Taylor, P. and Jonker, L. (1978). Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 40(1-2):145–156.
- Van Veelen, M. (2011). The replicator dynamics with n players and population structure. *Journal of Theoretical Biology*, 276(1):78–85.
- Wilson, D. and Colwell, R. (1981). Evolution of sex ratio in structured demes. *Evolution*, pages 882–897.

# An alternative route to robustness: The relationship between assortativity, in-components, and characteristic path length in gene regulatory networks

Dov A. Pechenick<sup>1</sup>, Joshua L. Payne<sup>2</sup>, and Jason H. Moore<sup>1</sup>

<sup>1</sup>Computational Genetics Laboratory, Dartmouth College, Hanover, New Hampshire, USA

<sup>2</sup>Institute of Evolutionary Biology and Environmental Studies, University of Zurich, Zurich, Switzerland  
Dov.A.Pechenick.GR@Dartmouth.edu

## Abstract

Gene regulatory networks (GRNs) comprise the interacting genes and gene products that drive genetic regulation within the cell. Because of the vital role they play in producing cell function, GRNs are robust to a variety of perturbations, including genetic mutation. There are multiple underlying causes for this robustness, including topological properties of GRNs, such as their degree distribution. Another topological property, assortativity, has recently been attributed to the robustness of GRNs. Assortative GRNs were found to have smaller in-components (ICs) than their disassortative counterparts, and this led to increased robustness to multiple types of genetic mutation. However, some assortative GRNs lacked the distinctive small ICs, yet were still robust. This suggests that assortativity affects robustness via multiple mechanisms, and unraveling these is a necessary step for understanding which specific features of GRNs give rise to their robustness. Here, we uncover a separate route by which assortativity affects robustness, whereby assortativity influences the characteristic path length of the GRN, which in turn alters robustness.

## Introduction

Gene expression produces the complex machinery necessary for cellular life, and its regulation is a crucial means by which cells can assume specific functions. For example, the regulation of gene expression enables cells to respond to different environments (Gasch et al., 2000; Causton et al., 2001) and navigate diverse paths of differentiation to produce distinct cell fates (Davidson, 2006; Huang et al., 2005).

One of the cell's implicit constructs that accomplishes this regulation is its network of gene-gene interactions, in which the product of one gene directly influences the expression of another gene, as happens with transcription factors. This network is referred to as a gene regulatory network (GRN). Because GRNs are often involved in critical biological functions, it is important that they are robust to genetic perturbation, such as gene knock-out (Jeong et al., 2001) or the rewiring of regulatory interactions (Isalan et al., 2008).

Several theoretical studies have attempted to elucidate the source of this robustness, and one source appears to be GRN topology. For example, GRNs that possess heavy-tailed degree distributions were shown to be more robust than those

that have other degree distributions (Aldana and Cluzel, 2003). This observation has been supported by empirical findings that suggest real-world GRNs have heavy-tailed degree distributions (Babu et al., 2004).

Assortativity is another topological property that has been shown to vary in real-world networks (Newman, 2002; Foster et al., 2010). The assortativity of a GRN measures the tendency for genes to interact with other similar genes. One measure of assortativity considers whether pairs of interacting genes have similar numbers of connections in the GRN, and this is referred to as degree assortativity. Recently, we presented theoretical work which showed that the increased degree assortativity of a GRN produces increased robustness to a variety of genetic perturbations, including point mutations (Pechenick et al., 2012) and gene birth (Pechenick et al., 2013). This occurs via a reduction in the average size of the in-components (ICs) of a GRN. An IC of a gene is the set of all other genes in the GRN which can directly or indirectly influence that gene's expression. We observed that IC sizes shrink with increasing assortativity, and showed how this explains the increased robustness (Pechenick et al., 2012). However, this mechanism did not explain all the observed changes to robustness, as some GRNs displayed increased robustness with increased assortativity but did not exhibit corresponding changes to their IC sizes. This observation invites further inquiry: What is the alternative route by which increased assortativity leads to increased robustness?

In this study, we uncover an additional mechanism whereby assortativity influences the robustness of GRNs. We first show that IC sizes do not always shrink with increasing assortativity, and that the GRNs with unaffected ICs are still more robust than their less assortative counterparts. We then show that while ICs are unchanged in these GRNs, their characteristic path length increases with increasing assortativity. Finally, we demonstrate that characteristic path length generally affects the robustness of GRNs.

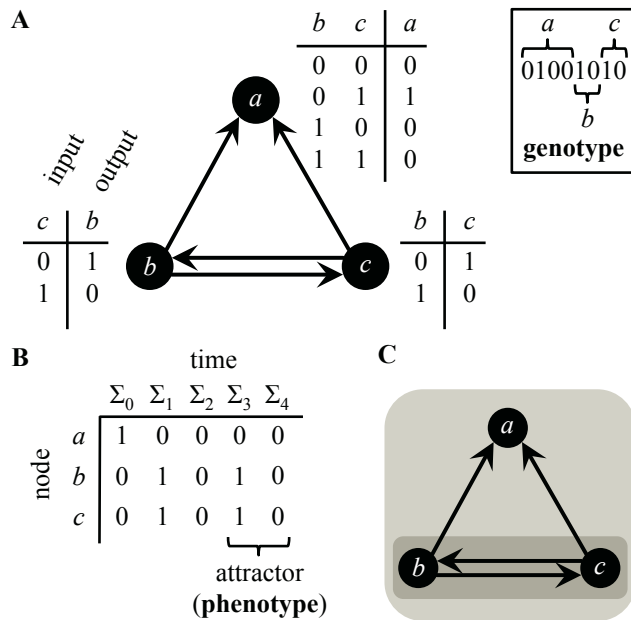


Figure 1: A small Boolean network example. (A) This network is composed of 3 nodes and 4 directed edges. Each node possesses a look-up table with the *cis*-regulatory logic that determines the dynamics of the Boolean network. This logic defines the expression state of the node at time  $t + 1$  as a function of the states of its inputs at time  $t$ . For example, the logic for node  $a$  shows how each possible combination of expression states  $\sigma_b(t)$  and  $\sigma_c(t)$  of the inputs at time  $t$  dictate the expression state  $\sigma_a(t + 1)$ . (A; box) The *cis*-regulatory logic for the entire network is its genotype. (B) Starting with the initial configuration  $\Sigma_0$  at time  $t = 0$ , the states are updated according to the genotype until they repeat, forming an attractor that is analogous to a phenotype. Here, the attractor length is two. (C) The in-component (IC) of node  $a$  is the set  $\{a, b, c\}$  (light grey), which includes  $a$  and all other nodes that directly or indirectly provide input to  $a$ . The ICs for  $b$  and  $c$  are both  $\{b, c\}$  (dark grey).

## Methods

### The Model

Random Boolean networks were used to computationally model genetic regulation (Kauffman, 1969), and will herein be referred to as gene regulatory networks (GRNs). In these GRNs, nodes represent genes and edges represent directed regulatory interactions between genes where the regulator regulates the target gene (Figure 1A). Binary gene expression is represented by the Boolean state of each gene, which is dictated at discrete time points by the Boolean functions that define the possible outcomes of the regulatory interactions. Boolean functions represent the *cis*-regulatory logic for each gene, and are commonly encoded as truth tables (Figure 1A). The state of a gene  $\sigma_i(t + 1)$  is determined by

a Boolean function  $f$  which considers the states of the  $k_{in}$  regulators of node  $i$  at time  $t$ :

$$\sigma_i(t + 1) = f_i(\sigma_{i_1}(t), \dots, \sigma_{i_{k_{in,i}}}(t)), \quad (1)$$

where  $\sigma_{i_1}, \dots, \sigma_{i_{k_{in,i}}}$  are the states of the  $k_{in,i}$  regulators of node  $i$ . This function is deterministic, and thus the states of all the genes that exist at the initial time point 0, referred to as configuration  $\Sigma_0$ , will invariably produce the subsequent configuration  $\Sigma_1$  at the next time point. In combination with the finite number of possible configurations ( $2^N$ , where  $N$  is the number of genes in the GRN), this determinism guarantees that once a configuration is encountered, a sequence of configurations will repeat indefinitely:

$$\Sigma_0 \rightarrow \dots \rightarrow \Sigma_t \rightarrow \dots \rightarrow \Sigma_{t+l-1} \rightarrow \Sigma_t \rightarrow \dots, \quad (2)$$

where  $l$  is the number of configurations in the repeated sequence, called an attractor of length  $l$  (Figure 1B). This represents a stable gene expression pattern for the GRN (Huang et al., 2005). Although general and abstract, these models have successfully recapitulated cellular responses in the yeast *Saccharomyces cerevisiae* (Serra et al., 2004), the fly *Drosophila melanogaster* (Albert and Othmer, 2003), the plant *Arabidopsis thaliana* (Espinosa-Soto et al., 2004), and the sea urchin *Strongylocentrotus purpuratus* (Peter et al., 2012).

### Genotype-to-Phenotype Mapping

In order to study how these GRNs maintain their function in the face of genetic perturbation, it is first necessary to explicitly define their mapping of genotype-to-phenotype. The *cis*-regulatory logic of the entire GRN was considered its genotype (Pechenick et al., 2012; Payne et al., 2013) (Figure 1A; box), as it dictates the overall dynamics of the GRN. The attractor that results from this genotype and some initial configuration  $\Sigma_0$  represents a phenotype of the GRN (Figure 1B) (Huang et al., 2005).

### Robustness

Upon establishing the genotype-to-phenotype mapping of these GRNs, the robustness of a phenotype to genetic perturbation must be defined in terms of a specific perturbation (Wagner, 2005). We considered point mutations to the genotype of the GRN, which represent mutations to the *cis*-regulatory regions of the genes in the GRN (Pechenick et al., 2012; Payne et al., 2013). The functional impact of such mutations has been demonstrated in a number of biological contexts (Wray, 2007), such as the patterning of bristles on the fly larvae *Drosophila sechellia* (Sucena and Stern, 2000), the skeletal development of the fish *Gasterosteus aculeatus* (Shapiro et al., 2004), and the branching structure of maize *Zea mays* (Clark et al., 2006).

To estimate robustness, a random walk was conducted in the genotype space of a GRN to determine the robust-

ness of the corresponding phenotype to genotypic perturbation. First, an initial configuration  $\Sigma_0$  and *cis*-regulatory logic were randomly selected for the GRN. This conservative approach eliminates any assumptions about the initial conditions of biological GRNs. Then, the genotype was subjected to a single bit flip. The attractor that resulted from  $\Sigma_0$  and the mutated *cis*-regulatory logic was then compared to the original attractor, and if they matched the mutation was considered neutral and the new *cis*-regulatory logic was preserved as the starting point for the next step in the random walk. Otherwise, the new *cis*-regulatory logic was discarded, and the next step was attempted using the *cis*-regulatory logic from the previous step. Upon completing the random walk, the proportion of mutations that were neutral was used as an estimate of robustness. This estimate approximates the average genotypic robustness for all genotypes that comprise a single phenotype, which is the definition of phenotypic robustness proposed by Wagner (2008).

### Construction of GRNs

GRNs with a heavy-tailed output degree distribution were constructed as follows: For each gene  $i$  in a GRN with  $N$  genes, its number of regulatory targets  $k_{\text{out},i}$  was selected from a power-law distribution (Darabos et al., 2009):

$$p(k_{\text{out}}) = \frac{1}{Z(\gamma)} k_{\text{out}}^{-\gamma}, \quad (3)$$

where  $Z(\gamma) = \sum_{j=1}^N j^{-\gamma}$  is the normalization constant. This generated an out-degree sequence  $k_{\text{out},1} \dots k_{\text{out},N}$ , which was used to randomly select the  $k_{\text{out},i}$  regulatory targets for each gene  $i$ . The resulting in-degree sequence  $k_{\text{in},1} \dots k_{\text{in},N}$  approximated a Poisson input degree distribution. The combination of Poisson input and power-law output degree distribution closely resembles empirical real-world GRN data, such as those from the microbes *E. Coli*, *B. Subtilis*, and *S. cerevisiae* (Aldana et al., 2007).

### Assortativity

Degree assortativity ( $r$ ), referred to here simply as assortativity, is a global network property that captures the tendency for nodes with similar degrees to share an edge between them. This property was defined by Newman (2002) and is calculated as

$$r = \frac{\frac{1}{M} \sum_i j_i k_i - (\frac{1}{M} \sum_i \frac{1}{2} (j_i + k_i))^2}{\frac{1}{M} \sum_i \frac{1}{2} (j_i^2 + k_i^2) - (\frac{1}{M} \sum_i \frac{1}{2} (j_i + k_i))^2}, \quad (4)$$

where  $M$  is the number of edges in the network,  $j_i$  and  $k_i$  are the degrees of the nodes at either end of edge  $i$ , and  $r$  resides in the domain  $[-1, 1]$ ;  $-1$  indicates maximum dissimilarity and  $1$  indicates maximum similarity between degrees of nodes that share an edge. In a directed network,  $j_i$  and  $k_i$  may each be one of two types of degree, in- and out-degree, which results in four possible types of assortativity (Foster et al., 2010). For the purposes of this study, only out-degree was considered, as in Pechenick et al. (2012, 2013).

### In-Components (ICs) and Characteristic Path Length

An in-component (IC) is a local network property. The IC of a node  $a$  corresponds to the set of nodes that are capable of influencing  $a$  (Figure 1C). The mean size of the ICs of all nodes in a network provides a measure of the extent to which nodes can affect other nodes in that network, and is calculated as

$$\bar{S} = \frac{\sum_{i=1}^N S_i}{N}, \quad (5)$$

where  $S_i$  is the IC size of node  $i$ , and  $N$  is the number of nodes in the network.

The characteristic path length is a global network property that also captures the relative ease with which information can flow between nodes in a network (Watts and Strogatz, 1998). This property is calculated by determining the shortest directed path between all pairs of nodes, and taking the mean of all existing paths. If a path does not exist between a pair of nodes, that pair is not considered in the calculation.

### GRN Rewiring

Upon construction, a GRN with  $N$  genes has an out-degree sequence  $k_{\text{out},1} \dots k_{\text{out},N}$  and an in-degree sequence  $k_{\text{in},1} \dots k_{\text{in},N}$ , and together these are referred to simply as its degree sequence. It is important to point out that different degree sequences can be drawn from the same degree distribution (Equation 3). In order to examine the effects of various topological properties on GRNs, it was desirable to vary those properties without altering the degree sequence of the GRN. An edge-swap algorithm was thus used to modify topology while keeping both the in- and out-degree of every gene intact (Milo et al., 2003). In each iteration of this algorithm, two edges  $i \rightarrow j$  and  $x \rightarrow y$  were selected, and the regulatory targets were swapped between the regulators to yield two new edges  $i \rightarrow y$  and  $x \rightarrow j$ . If the new edges caused the GRN to be closer to a desired value for a particular topological property, or if no change was observed with respect to this property, then the new edges were kept. Otherwise, the new edges were discarded and the old edges were kept.

### Simulation Design

To examine the relationship between assortativity, mean IC size, and characteristic path length, 2000 weakly connected GRNs with  $N = 30$  and  $\bar{k}_{\text{out}} = 4$  ( $\gamma = 1.55$ ) were constructed. Self-loops were excluded because they trivially increase assortativity without changing mean IC size or characteristic path length, and such exclusion did not affect past results (Pechenick et al., 2013). GRNs with  $\bar{k}_{\text{out}} = 4$  are in the chaotic dynamical regime, which was chosen because these GRNs tend to have large ICs and exhibit dramatic variation in robustness, whereas ordered and critical GRNs exhibit limited changes in robustness as assortativity varies (Pechenick et al., 2012, 2013). The 2000 degree sequences



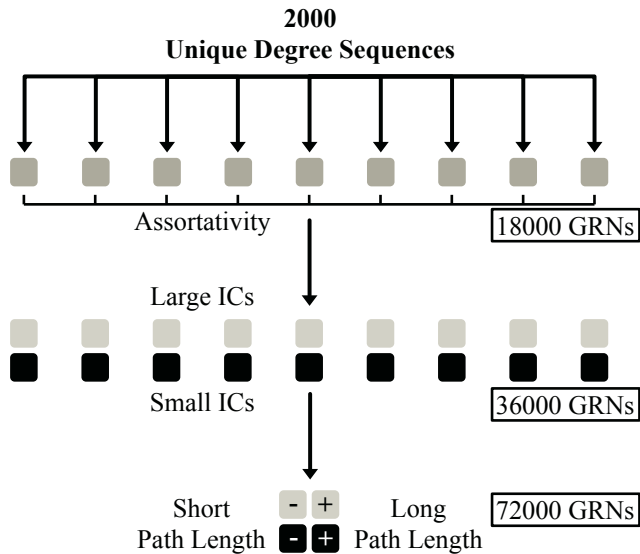


Figure 2: Simulation design flowchart. Beginning with 2000 unique degree sequences, every degree sequence was rewired (preserving degree sequence) to construct GRNs at each of 9 evenly spaced assortativity values. Each GRN was then rewired (preserving degree sequence and assortativity) to produce two new GRNs, one with small and one with large mean IC size. Next, each of those GRNs was rewired (preserving degree sequence, assortativity, and mean IC size), producing one GRN with short and one with long characteristic path length. This final step is only displayed once for clarity.

were then rewired to 9 evenly spaced assortativity values in the range  $[-0.64, -0.02] \pm 0.01$  (Figure 2), where the edge-swap algorithm proceeded until assortativity was within 0.01 of the desired value, as in Pechenick et al. (2013). This resulted in 18000 GRNs, where every degree sequence was represented at every assortativity value. Note that heavy-tailed degree distributions are inherently negatively assortative (Johnson et al., 2010), and the domain of the assortativity values considered here was entirely negative.

To isolate the effects of mean IC size, GRNs were rewired to low and high values for mean IC size. The edge-swap algorithm was allowed to minimize or maximize mean IC size for each GRN until no desired change in mean IC size had been observed for 10000 iterations, and assortativity was constrained within  $\pm 0.02$  of the original 9 evenly spaced values. Here, every degree sequence was represented twice at every assortativity value. We refer to these two classes of GRNs as “small ICs” and “large ICs.”

Then, to isolate the effects of characteristic path length for each of the two classes of GRNs, GRNs were rewired to low and high values for characteristic path length. As before, the edge-swap algorithm was allowed to proceed until the

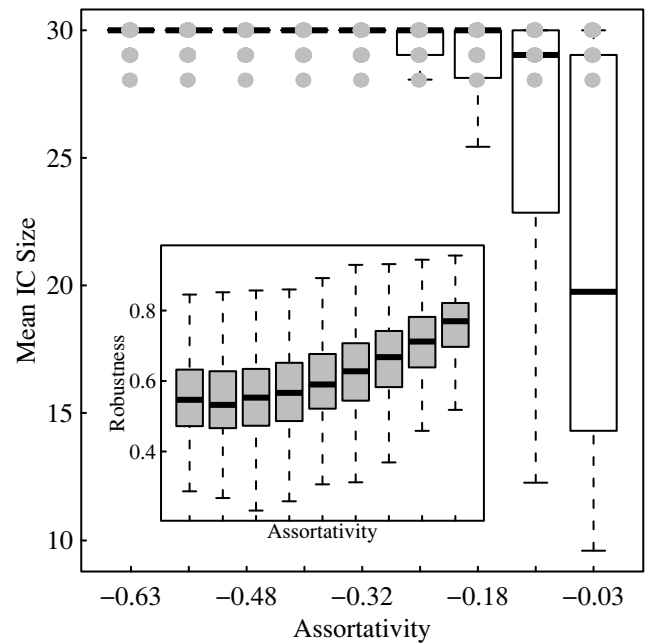


Figure 3: Mean in-component (IC) size vs. assortativity. The mean IC size of the 18000 GRNs decreases as assortativity increases ( $p = 0.025$ , Spearman’s rank correlation on median values). However, 422 (21%) of the 2000 degree sequences do not exhibit a decrease in mean IC size (grey circles). (Inset) Robustness vs. assortativity. Robustness for the 422 degree sequences (3798 GRNs) increases as assortativity increases ( $p \ll 0.001$ , Spearman’s rank correlation on median values). Outliers are omitted from plots for clarity.

desired decrease or increase was not observed for 10000 iterations. Assortativity was constrained within  $\pm 0.02$ , as before, and mean IC size was not allowed to vary at all from its starting value. Here, every degree sequence was represented four times at every assortativity value for every combination of low and high mean IC size and characteristic path length. For more on rewiring networks to obtain multiple desired topological properties, see Holme and Zhao (2007).

The robustness of a GRN was estimated by taking the average of the outcomes for 100 random walks with different initial configurations  $\Sigma_0$  and *cis*-regulatory logic, where each random walk consisted of 500 attempted steps. These parameters were selected as a compromise between accuracy and computational efficiency (Pechenick et al., 2012).

## Results

### *Small mean IC size is not solely responsible for increased robustness.*

Consistent with previous observations (Pechenick et al., 2012, 2013), the mean IC sizes of the 18000 GRNs at 9 assortativity values tended to decrease with increasing assortativity (Figure 3). This decrease in mean IC size was impli-

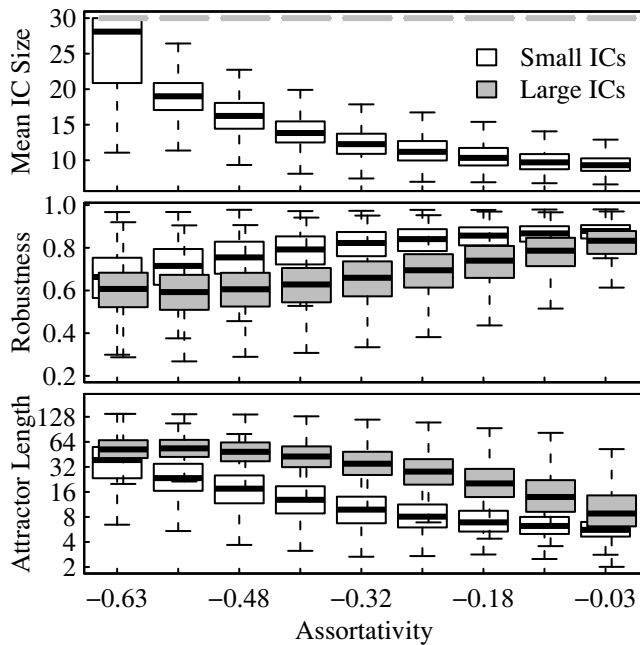


Figure 4: Mean in-component (IC) size, robustness, and attractor length vs. assortativity for GRNs with either small or large ICs. At every assortativity value, (top) GRNs with large ICs have significantly larger mean IC size, (middle) lower robustness, and (bottom) longer attractors than GRNs with small ICs (all  $p \ll 0.001$ , Wilcoxon rank sum test). Outliers are omitted from plots for clarity.

cated in a corresponding increase in robustness (Pechenick et al., 2012); however, some degree sequences produced GRNs with identical mean IC sizes at every assortativity value (Figure 3; grey circles), and these GRNs also showed increased robustness with increased assortativity (Figure 3; inset). Therefore, a topological mechanism distinct from mean IC size is required to explain the observed increase in robustness for these GRNs.

### ***Both small and large mean IC sizes are possible for the same degree sequences.***

One possible explanation for the two types of GRNs observed in Figure 3 is that certain degree sequences are simply incapable of rewiring in such a way that results in smaller ICs. Likewise, some degree sequences may be forced to form smaller ICs as assortativity increases. If this were true, not only would multiple mechanisms be needed to explain the observed increases in robustness, but these mechanisms would act exclusively on certain degree sequences and not others.

In order to test whether this is indeed the case, we rewired each degree sequence at each assortativity value to try to obtain two new GRNs with a small and large mean IC size (preserving assortativity), respectively (Figure 4; top). At the

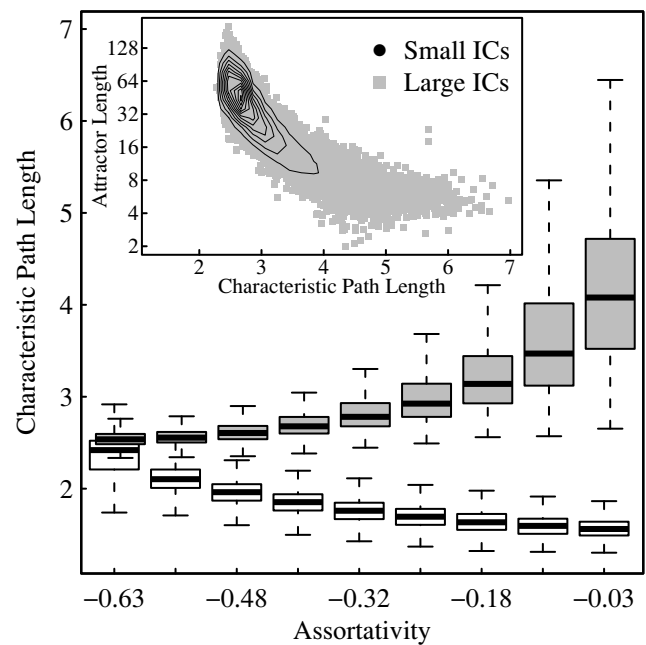


Figure 5: Characteristic path length vs. assortativity. The characteristic path length of GRNs with large ICs increases as assortativity increases, whereas it decreases for GRNs with small ICs (both  $p \ll 0.001$ , Spearman's rank correlation on median values). Outliers are omitted from the plot for clarity. (Inset) Attractor length vs. characteristic path length. The attractor length of GRNs with large ICs decreases as characteristic path length increases, whereas it increases for GRNs with small ICs (both  $p \ll 0.001$ , Spearman's rank correlation on all values). Contour lines are provided as a visual guide for the relative density of points.

lowest assortativity value, where GRNs almost exclusively have large ICs, 47% of degree sequences failed to produce GRNs with smaller ICs. However, this dropped to 13% at the second lowest assortativity value, 2% at the third, and  $< 1\%$  for all other assortativity values. For the top three assortativity values, every degree sequence was able to produce distinct GRNs with either small or large ICs. Therefore, while multiple mechanisms are still necessary to understand how assortativity influences robustness, these respective mechanisms are not restricted to only certain degree sequences. We then estimated robustness for these two classes of GRNs and found that for every assortativity value GRNs with small ICs were significantly more robust than their counterparts with large ICs (Figure 4; middle). This suggests that the previously described mechanism for how small ICs can lead to increased robustness (Pechenick et al., 2012) is generally relevant for GRNs across different degree sequences and at a wide range of assortativity values.

However, since GRNs with large ICs displayed increased robustness with increased assortativity (Figure 4; middle),

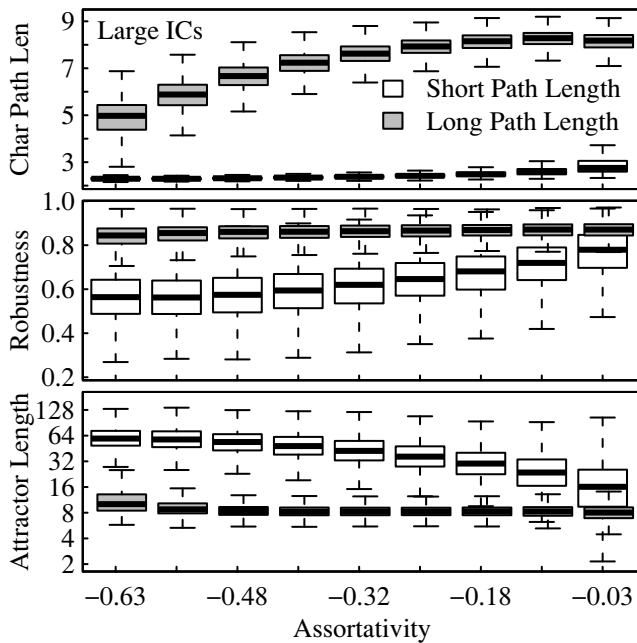


Figure 6: For GRNs with large ICs: Characteristic path length, robustness, and attractor length vs. assortativity for GRNs with either short or long characteristic path length. At every assortativity value, (top) GRNs with long characteristic path length have significantly longer paths, (middle) higher robustness, (bottom) and shorter attractors than GRNs with short characteristic path length (all  $p \ll 0.001$ , Wilcoxon rank sum test). Outliers are omitted from plots for clarity.

another general mechanism besides mean IC size is required to explain the effect assortativity has on robustness. In Pechenick et al. (2012), we proposed that mean IC size affects robustness by altering the attractor lengths of GRNs, and we observed here that attractor length decreases with assortativity for both classes of GRNs (Figure 4; bottom). This is consistent with a negative relationship between robustness and attractor length, and suggests that the alternative mechanism producing highly robust GRNs with large ICs is doing so by reducing attractor length.

#### ***Characteristic path length changes with assortativity.***

For the GRNs with large ICs, mean IC size does not change with assortativity (Figure 4; top), so another topological property must be influencing the attractor lengths of these GRNs. We found that the characteristic path length of GRNs changes with assortativity in a manner that is dependent on which of the two GRN classes is being considered (Figure 5). For GRNs with small ICs, characteristic path length shrinks with increasing assortativity, and appears to be associated with a shrinking mean IC size (Figure 4; top). However, for GRNs with large ICs, characteristic path length

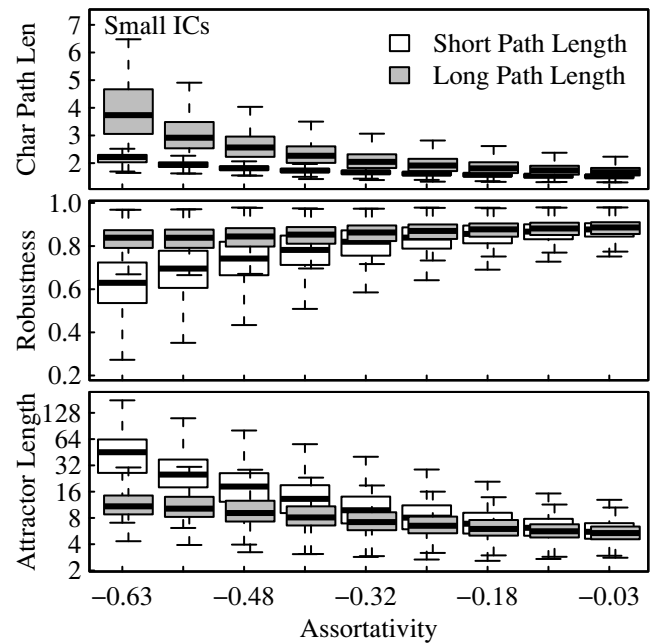


Figure 7: For GRNs with small ICs: Characteristic path length, robustness, and attractor length vs. assortativity for GRNs with either short or long characteristic path length. At every assortativity value, GRNs with long characteristic path length are significantly different from GRNs with short characteristic path length, as in Figure 6 (all  $p \ll 0.001$ , Wilcoxon rank sum test). Outliers are omitted from plots for clarity.

grows with increasing assortativity. These opposing trends are accompanied by two additional opposing trends: For GRNs with small ICs, attractor length is positively correlated with characteristic path length, whereas for GRNs with large ICs, attractor length is negatively correlated with characteristic path length (Figure 5; inset). This is consistent with the decreases in attractor length as assortativity increases for both GRNs with small and large ICs (Figure 4; bottom).

#### ***Long characteristic path length contributes to robustness.***

To determine whether characteristic path length is directly responsible for changes in robustness and attractor length, we varied the characteristic path length of GRNs to their high and low bounds while preserving both assortativity and mean IC size for GRNs with large ICs (Figure 6) and small ICs (Figure 7). For GRNs with large ICs, the maximum characteristic path length that was achievable increased with increasing assortativity (Figure 6; top), which echoes the relationship between these two properties observed in Figure 5. However, these results go on to show that GRNs with longer characteristic path length have higher robustness

(Figure 6; middle) and shorter attractors (Figure 6; bottom) than their counterparts with shorter characteristic path length that possess the same assortativity and mean IC size. These results argue for a direct positive role for characteristic path length in affecting the robustness of GRNs with large ICs at a wide range of assortativity values.

For GRNs with small ICs, although changes in characteristic path length seemed to have the opposite effect on attractor length, and therefore robustness, as GRNs with large ICs (Figure 5; inset), it was unclear whether this represented a direct relationship between the two properties. In particular, for GRNs with small ICs, increasing assortativity produces decreases in both characteristic path length (Figure 5) and mean IC size (Figure 4; top). Since a decrease in mean IC size leads to increased robustness (Figure 4; middle), it is difficult to disentangle the effects of characteristic path length from those caused by changes in mean IC size. By varying characteristic path length independently of both assortativity and mean IC size, we can directly address this.

In contrast to GRNs with large ICs, for GRNs with small ICs, the maximum characteristic path length that was achievable decreased with increasing assortativity (Figure 7; top), likely reflecting a positive association between characteristic path length and mean IC size. However, consistent with the results for GRNs with large ICs, GRNs with small ICs that possess long characteristic path length exhibit higher robustness (Figure 7; middle) and shorter attractors (Figure 7; bottom) than their counterparts with short characteristic path length with the same assortativity and mean IC size. Taken together, these results indicate that although the ability to vary the characteristic path length of a GRN depends on both its mean IC size and its assortativity, adopting a longer characteristic path length leads to higher robustness in a manner that is independent of these other two properties. Therefore, as the assortativity of a GRN increases, one of two things tends to occur that can result in higher robustness. Its mean IC size may shrink, which leads to higher robustness in a manner that dominates the effects of the associated shrinking of characteristic path length. Or, its mean IC size may not shrink, in which case characteristic path length will tend to grow and lead to higher robustness.

## Discussion

We have presented an alternative mechanism by which assortativity influences the robustness of GRNs to mutations in their *cis*-regulatory logic. It is often the case that an increase in assortativity results in a decrease in mean IC size, which increases the robustness of the GRN. However, even when mean IC size does not change, robustness nonetheless increases. We have found that in this case, an increase in assortativity leads to an increase in characteristic path length, which is associated with increased robustness. Furthermore, this effect was not limited to GRNs with large mean IC sizes. The assortativity and mean IC size of a GRN does constrain

its characteristic path length. Nevertheless, we have shown that a GRN with a long characteristic path length is on average more robust than a GRN with similar assortativity and mean IC size, but with a shorter characteristic path length.

These results complement previous theoretical work that showed that the characteristic path length of network models influences their dynamics. In contrast to the inverse relationship between characteristic path length and attractor length that we observed, Serra and Villani (2002) showed that decreasing the characteristic path length of cellular automata (CA) led to simpler dynamics. This result is likely due to their use of the majority update rule, which took advantage of the shorter paths to more easily achieve uniform behavior across the network. In line with what we have shown, Lizier et al. (2011) observed that an increase in the characteristic path length of random Boolean networks led to greater information storage and less information transfer, which are properties that they found associated with the simpler dynamics typically found in the ordered dynamical regime.

As we gather more data about the structure of biological GRNs, the results presented in this work will provide a theoretical basis for searching for specific topological features that are indicative of robustness. Indeed, high assortativity would imply robustness, yet its absence would not discount it. A relatively small mean IC size could suggest robustness at a range of assortativity values, and yet it too is not exclusively necessary. In the absence of a small mean IC size, a long characteristic path length would signal robustness that could rival the robustness of a GRN that did possess a small mean IC size. We have shown that any one or a combination of these properties contributes to highly robust GRNs. Furthermore, this study exclusively considered out-degree assortativity, and further work will be necessary to determine how the other types of degree assortativity are involved (Foster et al., 2010). As we continue to map and examine biological GRNs, it will be informative to catalog their topological properties in an attempt to understand whether they depict common or varied evolutionary strategies of achieving robustness.

## Acknowledgements

This work was supported by NIH grant Nos. R01 EY022300, R01 LM009012, R01 LM010098 and R01 AI59694. D.A.P. was supported by Award Number T32GM008704 from the National Institute of General Medical Sciences. J.L.P. was supported by a Collaborative Research Travel Grant from the Burroughs Wellcome Fund and an International Research Fellowship from the National Science Foundation.

## References

- Albert, R. and Othmer, H. G. (2003). The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *Journal of Theoretical Biology*, 223:1–18.



- Aldana, M., Balleza, E., Kauffman, S., and Resendiz, O. (2007). Robustness and evolvability in genetic regulatory networks. *Journal of Theoretical Biology*, 245(3):433–448.
- Aldana, M. and Cluzel, P. (2003). A natural class of robust networks. *Proceedings of the National Academy of Sciences USA*, 100(15):8710–8714.
- Babu, M., Luscombe, N., Aravind, L., Gerstein, M., and Teichmann, S. (2004). Structure and evolution of transcriptional regulatory networks. *Current Opinion in Structural Biology*, 14(3):283–291.
- Causton, H. C., Ren, B., Koh, S. S., Harbison, C. T., Kanin, E., Jennings, E. G., Lee, T. I., True, H. L., Lander, E. S., and Young, R. A. (2001). Remodeling of yeast genome expression in response to environmental changes. *Molecular Biology of the Cell*, 12:323–337.
- Clark, R. M., Wagler, T. N., Quijada, P., and Doebley, J. (2006). A distant upstream enhancer at the maize domestication gene *tb1* has pleiotropic effects on plant and inflorescent architecture. *Nature Genetics*, 38(5):594–597.
- Darabos, C., Tomassini, M., and Giacobini, M. (2009). Dynamics of unperturbed and noisy generalized Boolean networks. *Journal of Theoretical Biology*, 260(4):531–544.
- Davidson, E. H. (2006). *The Regulatory Genome: Gene Regulatory Networks in Development and Evolution*. Academic Press.
- Espinosa-Soto, C., Padilla-Longoria, P., and Alvarez-Buylla, E. R. (2004). A gene regulatory network model for cell-fate determination during *Arabidopsis thaliana* flower development that is robust and recovers experimental gene expression profiles. *The Plant Cell*, 16:2923–2939.
- Foster, J., Foster, D., Grassberger, P., and Paczuski, M. (2010). Edge direction and the structure of networks. *Proceedings of the National Academy of Sciences*, 107(24):10815–10820.
- Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., Botstein, D., and Brown, P. O. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11:4241–4257.
- Holme, P. and Zhao, J. (2007). Exploring the assortativity-clustering space of a network's degree sequence. *Physical Review E*, 75(4):046111.
- Huang, S., Eichler, G., Bar-Yam, Y., and Ingber, D. (2005). Cell fates as high-dimensional attractor states of a complex gene regulatory network. *Physical Review Letters*, 94(12):128701.
- Isalan, M., Lemerle, C., Michalodimitrakis, K., Horn, C., Beltrao, P., Raineri, E., Garriga-Canut, M., and Serrano, L. (2008). Evolvability and hierarchy in rewired bacterial gene networks. *Nature*, 452(7189):840–845.
- Jeong, H., Mason, S. P., Barabási, A.-L., and Oltvai, Z. (2001). Lethality and centrality in protein networks. *Nature*, 411:41–42.
- Johnson, S., Torres, J. J., Marro, J., and Muñoz, M. A. (2010). Entropic origin of disassortativity in complex networks. *Physical Review Letters*, 104(10):108702.
- Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467.
- Lizier, J. T., Pritam, S., and Prokopenko, M. (2011). Information dynamics in small-world boolean networks. *Artificial Life*, 17(4):293–314.
- Milo, R., Kashtan, N., Itzkovitz, S., Newman, M., and Alon, U. (2003). On the uniform generation of random graphs with prescribed degree sequences. *Arxiv preprint cond-mat/0312028*.
- Newman, M. (2002). Assortative mixing in networks. *Physical Review Letters*, 89(20):208701.
- Payne, J. L., Moore, J. H., and Wagner, A. (2013). Robustness, evolvability, and the logic of genetic regulation. *Artificial Life*, Special Issue for the 20<sup>th</sup> Anniversary of the European Conference on Artificial Life (ECAL 2011), In Press.
- Pechenick, D. A., Moore, J. H., and Payne, J. L. (2013). The influence of assortativity on the robustness and evolvability of gene regulatory networks upon gene birth. *Journal of Theoretical Biology*, 330:26–36.
- Pechenick, D. A., Payne, J. L., and Moore, J. H. (2012). The influence of assortativity on the robustness of signal-integration logic in gene regulatory networks. *Journal of Theoretical Biology*, 296:21–32.
- Peter, I. S., Faure, E., and Davidson, E. H. (2012). Predictive computation of genomic logic processing functions in embryonic development. *Proceedings of the National Academy of Sciences*, 109(41):16434–16442.
- Serra, R. and Villani, M. (2002). Perturbing the regular topology of cellular automata: implications for the dynamics. In *Cellular Automata*, pages 168–177. Springer.
- Serra, R., Villani, M., and Semeria, A. (2004). Genetic network models and statistical properties of gene expression data in knock-out experiments. *Journal of Theoretical Biology*, 227:149–157.
- Shapiro, M. D., Marks, M. E., Peichel, C. L., Blackman, B. K., Nereng, K. S., Jónsson, B., Schluter, D., and Kingsley, D. M. (2004). Genetic and developmental basis of evolutionary pelvic reduction in threespine sticklebacks. *Nature*, 428(6984):717–723.
- Sucena, É. and Stern, D. L. (2000). Divergence of larval morphology between *Drosophila sechellia* and its sibling species caused by cis-regulatory evolution of *ovo/shaven-baby*. *Proceedings of the National Academy of Sciences*, 97(9):4530–4534.
- Wagner, A. (2005). *Robustness and Evolvability in Living Systems*. Princeton University Press.
- Wagner, A. (2008). Robustness and evolvability: a paradox resolved. *Proceedings of the Royal Society B*, 275(1630):91–100.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature*, 393(6684):440–442.
- Wray, G. A. (2007). The evolutionary significance of cis-regulatory mutations. *Nature Reviews Genetics*, 8(3):206–216.

## The detection of intermediate-level emergent structures and patterns

M. Villani<sup>1,2</sup>, A. Filisetti<sup>1,3</sup>, S. Benedettini<sup>1</sup>, A. Roli<sup>1,4</sup>, D. Lane<sup>1,5</sup> and R. Serra<sup>1,2</sup>

<sup>1</sup> European Centre for Living Technology, Ca' Minich, S. Marco 2940, 30124 Venezia, Italy

<sup>2</sup> Dept. of Physics, Informatics and Mathematics, University of Modena e Reggio Emilia, v. Campi 213b, 41125 Modena, Italy

<sup>3</sup> Energy and Environment Interdepartmental Center for Industrial Research, University of Bologna, Italy

<sup>4</sup> DISI Alma Mater Studiorum University of Bologna Campus of Cesena, Via Venezia 52, I-47521 Cesena, Italy

<sup>5</sup> Dept. of Communication and Economics, University of Modena e Reggio Emilia, v. Allegri 9, 41121 Reggio Emilia, Italy

marco.villani@unimore.it

### Abstract

Artificial life is largely concerned with systems that exhibit different emergent phenomena; yet, the identification of emergent structures is frequently a difficult challenge. In this paper we introduced a system to identify candidate emergent mesolevel dynamical structures in dynamical networks. This method is based on an extension of a measure introduced for detecting clusters in biological neural networks; its main novelty in comparison to previous application of similar measures is that we used it to consider truly dynamical networks, and not only fluctuations around stable asymptotic states. The identified structures are clusters of elements that behave in a coherent and coordinated way and that loosely interact with the remainder of the system. We have evidence that our approach is able to identify these “emerging things” in some artificial network models and in more complex data coming from catalytic reaction networks and biological gene regulatory systems (*A.thaliana*). We think that this system could suggest interesting new ways in dealing with artificial and biological systems.

### Introduction

Artificial life is largely concerned with systems that exhibit different emergent phenomena, life itself being one of the most intriguing ones. Yet defining emergence is a controversial issue, since it is deeply related to the relationship between the observer and the observed system. We will not enter here this debate, but we rather want to stress an aspect of emergence that is often overlooked, i.e. its intermediate-level characteristics.

Most discussions of emergence, as well as its existing theories and models, take into account a two-level system, and describe the bottom-up features of the phenomenon. For example, take the well-known Benard-Marangoni hexagonal convection pattern (Haken H. 2004) that is generated when the heat flow exceeds a certain threshold: here the microscopic level is that of the water “particles” and the macroscopic one is that of the hexagonal convection cells (in this case, the hierarchy of levels is related to their characteristic dimension). There is indeed a further upper

level, i.e. that of the apparatus where the phenomenon takes place; this uppermost level is necessary, and indeed it determines some major features of the phenomenon, as it can be seen e.g. by replacing the free surface with a metallic plate, thereby changing the pattern from hexagonal cells to cylindrical rolls. However the uppermost level is not affected by what happens at the lower levels and it therefore just provides the fixed boundary conditions that allow the establishment of the emergent patterns.

However, at a close look one finds that most emergent phenomena take place at levels that can be regraded as intermediate between pre-existing levels, that are in turn affected by the appearance of the intermediate emergent pattern. This topic is strictly related to the concept of emergence of hierarchies (Salthe, 1985) (Emmeche et al, 1997). Here we focus on the so-called “sandwiched” emergent phenomena, which appear in several fields such as physics, biology and social science (Lane et al, 2009). The most striking case is likely to be that of the formation of organs and tissues in multicellular organisms. Multicellularity predates the formation of organs, so the microscopic and macroscopic levels, i.e. cells and organism, were already in place when organs appeared. However, once they were formed, both organisms and cells were modified. Other examples of sandwiched emergence include the formation of clouds in physics and that of political factions, within parties, in social science, but there are actually very many. Indeed, once the importance of mesolevel emergence has been appreciated, it becomes difficult to find truly two-level systems in the sense defined above.

While in some cases it may be simple to identify the emergent structures or patterns, this is not always the case. Take for example a network of nodes that lacks any explicit all-encompassing spatial regularity, like e.g. a model of a genetic regulatory network with random connections, or a random chemical reaction network. While in spatially regular systems the appearance of regular patterns (like in the Benard case) or of clusters of nodes may be easy to find, in random systems that is by far more difficult.

In real genetic networks a lot of effort has been devoted to identifying frequently occurring motifs, i.e. small connection patterns that are much more frequent than what might be

expected if the network had been completely random; their high relative frequency can be regarded as a hint to the fact that they might have been selected by evolution due to the usefulness of the functions they perform. Indeed, the search for relevant connection patterns in complex networks is an important research topic. However, these approaches are mainly concerned with features that are directly related to the network topology, while here we want to look for structures and patterns that can be observed while looking at the dynamics of the system.

So, in order to escape from a merely topological view, we consider different subsets of the system, looking for those whose elements appear to be well coordinated among themselves and have a weaker interaction with the rest (Mesolevel Dynamical Structures, or MDS, in the following). For each subset of elements we will measure its so-called cluster index, a measure based on information theory that has been proposed by Tononi and Edelman (Tononi et al. 1998). After a suitable normalization procedure we rank the various subsets in order to identify those that are good candidates for the role of partially independent "organs" (note that they not necessarily exist in any network).

### The approach

For the sake of definiteness, let us consider a system  $U$ , our "universe" that is a network of  $N$  elements that can change their state in discrete time, taking one of a finite number  $l$  of discrete values. The value of element  $i$  at time  $t+1$ ,  $x_i(t+1)$ , will depend in a deterministic way upon the values of a fixed set of input elements at time  $t$ , possibly including the  $i$ -th (self-loops are not prohibited).

We will consider the systems' behaviors after an adequate relaxation time, in order to observe its asymptotic states. Given this quasi-equilibrium hypothesis we can estimate the entropy of each element from a long series of states by taking its frequencies  $f_v$  of observed values as proxies for probabilities, so:

$$H_i = -\sum_{v=1}^m f_v \log f_v \quad [1]$$

where the sum is taken over all the possible values an element can take. Of course, the average entropy of the whole system is the average of  $H_i$  taken over all the elements.

In case of a fixed point attractor  $H_i=0$  for every element since each node takes its value with frequency one. In order to apply entropy-based methods, Edelman and Tononi considered a system subject to gaussian noise around an equilibrium point. However nonlinear systems can carry several different attractors, each attractor revealing a particular way of functioning of the system itself: so the composition of all these asymptotic behaviors should help us in finding the parts of the system able to dynamically support them. Our "long data series" therefore will be composed by several repetitions of a single attractor, followed by repetitions of another one, etc. (ignoring the short transients

between the attractors)<sup>1</sup>, the number of repetitions reflecting the nature of the system we are analyzing. There are several different strategies to estimate these attractors' weights: in case of noisy systems a possibility is that of using the persistence time of the systems in each of them (Villani and Serra, 2013), whereas deterministic systems might be analyzed by weighting attractors with their basins of attraction. Given the nature of the cases of this work in the following we opt here for this second choice.

Now let us look for interesting sets of nodes (clusters, from now on). A good cluster should be composed by nodes (i) that possess high integration among themselves and (ii) that are more loosely coupled to other nodes of the system. The measure we define, called the cluster index, provides a value that can be used to rank various candidate clusters (i.e., emergent intermediate-level sets of coordinated nodes).

### The cluster index

Following Edelman and Tononi (Tononi et al. 1998), we will define the cluster index  $C(S)$  of a set  $S$  of  $k$  elements, as the ratio of a measure of their integration  $I(S)$  to a measure of the mutual information  $M(S;U-S)$  of that cluster with the rest of the system.

The integration is defined as follows: let  $H(S)$  be the entropy (computed as before) of the elements of  $S$ . This means that each state is a vector of  $k$  elements, and that the entropies are computed by counting the frequencies of the  $k$ -dimensional vectors. Then:

$$I(S) = \sum_{j \in S} H(x_j) - H(S) \quad [2]$$

So  $I(S)$  measures the deviation from statistical independence of the  $k$  elements in  $S$ , by subtracting the entropy of the whole subset to the sum of the single-node entropies. The mutual information of  $S$  to the rest of the world  $U-S$  is also defined by:

$$\begin{aligned} M(S;U-S) &\equiv H(S) + H(S|U-S) = \\ &= H(S) + H(U-S) - H(S,U-S) \end{aligned} \quad [3]$$

where, as usual,  $H(A|B)$  is the conditional entropy and  $H(A,B)$  the joint entropy. Finally, the cluster index  $C(S)$  is defined by:

$$C(S) = \frac{I(S)}{M(S;U-S)} \quad [4]$$

The cluster index vanishes if  $I=0$ ,  $M \neq 0$ , and is not defined whenever  $M=0$ . These cases, in which  $S$  is statistically independent from the rest of the system, can nevertheless be diagnosed in advance: the  $0/0$  form does not provide any information, whereas  $I(S)/0$  form - with  $I(S) \neq 0$  - points to statistical independence of  $S$  from the rest of the system, and calls for a separate analysis.

<sup>1</sup> Note that - given the nature of the average computation - the particular order of the data vectors on the series do not alter the analysis; in addition the data series can be composed by states belonging to different attractors

$C(S)$  scales with the size of the subsystem, so a loosely connected subsystem may have a larger index than a more coherent, smaller one: to compare the indices of the various candidate clusters it is therefore necessary to normalize their cluster indexes, for example by comparing them with those of subsystems having same size, but belonging to a non-clustered homogeneous system (a “null system”).

The definition of the “null system” is critical: it could be problem-specific, but we prefer a simple solution which is fairly general: given a series of discrete vectors, we compute the frequency of each symbol and generate a new random series where each symbol has a probability of appearing equal to that of the original series. This random null hypothesis is easy to calculate, related to the original data and parameter-free; moreover it satisfies the requirements set by Tononi of homogeneity and cluster-freeness.

The “null system” therefore provide us with a null hypothesis and allows us to calculate a set of normalization constants, one for each subsystem size. For each subsystem size, we compute average integration  $\langle I_h \rangle$  and mutual information  $\langle M_h \rangle$  (subscript  $h$  stands for “homogeneous”); we can then normalize the cluster index value of any subsystem  $S$  using the appropriate normalization constants dependent on the size of  $S$ :

$$C'(S) = \frac{I(S)}{\langle I_h \rangle} \bigg/ \frac{M(S; U-S)}{\langle M_h \rangle} \quad [5]$$

In order to compute a statistical significance index ( $T_c$  in the following) we apply this normalization to both the cluster indexes in the analyzed system and in the null system:

$$T_c(S) = \frac{C'(S) - \langle C'_h \rangle}{\sigma(C'_h)} \quad [6]$$

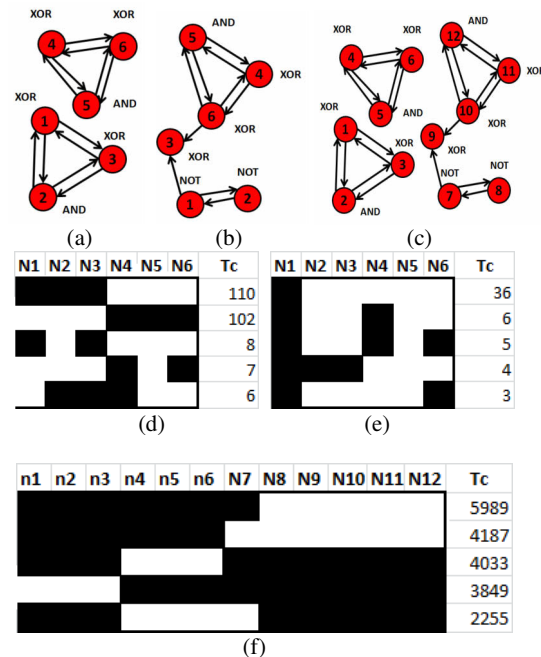
where  $\langle C'_h \rangle$  and  $\sigma(C'_h)$  are respectively the average and the standard deviation of the population of normalized cluster indices with the same size of  $S$  from the null system (Benedettini 2013). Finally we use  $T_c$  to rank the obtained clusters.

## Results

The cluster index has been introduced by Tononi (Tononi et al. 1998) for quasi-static systems; in the previous section we have shown how it could be extended to nonlinear dynamical systems, and in the following we will show the result of the application of this ranking method to some relevant systems, including generic models of gene regulatory networks, models of sets of catalytic chemical reactions and models of specific regulatory networks (*A.thaliana*). The method draws our attention on the subsets of the analyzed system that are highly functionally correlated and that could represent possible candidates MDSs. In the end we will also comment on the fact that our method, although not yet fully developed, outperforms usual correlation techniques.

## Boolean networks

The case study we are going to examine consists of three synchronous deterministic Boolean networks (BNs), described in Fig.1. BNs are an important framework frequently used to model genetic regulatory networks (Kauffman, 1993) (Kauffman, 1995), also applied to relevant biological data (Serra et al. 2004) (Shmulevich et al. 2005) (Villani et al. 2007) and processes (Serra et al. 2010) (Villani et al. 2011). The aim of this case study is to check whether CI analysis is capable of recognizing special topological cases, such as causally (in)dependent subnetworks and oscillators, where the causal relationships are more than binary. Note that given this “more than binary” nature in all the following cases, traditional analyses based on correlation between pairs of variables might fail. For example the computation of Pearson correlation coefficients of the networks of this section does not lead to identify related variables, given that only diagonal elements take non negligible values.



**Figure 1** (a) independent Boolean networks (BN1); (b) interdependent networks (BN2); (c) a system composed by the merging of both the previous networks (BN3). Beside each boolean node there is the boolean function the node is realizing. The second part of the figure shows the matrices illustrating the elements belonging to the clusters (white on figures) and the corresponding  $T_c$  values, for (d) BN1, (e) BN2 and (f) BN3 systems

CI analysis is able to correctly identify the two subnetworks of BN1 (first and second rows). The analysis clusters together 5 of 6 nodes of BN2: those already clustered in BN1, plus nodes 1 and 2 (which negates each other - figure 1b) and the node that compute the XOR of the signal coming from the two just mentioned groups. Indeed, all these nodes are needed in order to correctly reconstruct the BN2 series. The analysis



is able to identify all MDSs also when all the series are merged together (figure 1f, where the top two clusters correspond respectively to the 5 nodes already recognized in BN2 and to the whole BN2 system, while the third and fourth rows correspond to the independent subgraphs of BN1 - see (Villani et al., 2013) for details). Experiments performed using asynchronous update yielded essentially the same results with respect to both CI and correlation analyses.

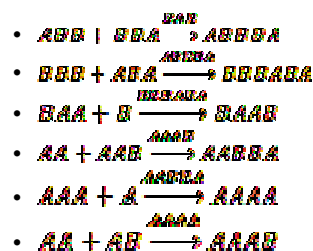
We would like to point out that CI analysis does not require any knowledge about system topology or dynamics. This information is normally unavailable in real cases; on the other hand, our methodology just needs a data series.

### Perturbing a catalytic reactions system

It is widely believed that the origin of life required the formation of sets of molecules able to collectively self-replicate (Carletti et al., 2008) (Filisetti et al., 2010) (Ganti, 2003) (Luisi et al., 2006) (Mansy et al., 2008) (Rasmussen et al., 2003) (Stano and Luisi, 2010) (Szostak et al., 2001), a phenomenon that may play an important role also in future bio-technological systems (Solè et al., 2007). There are many efforts to identify the dynamical cores of these systems, mainly based on static properties of the reaction networks they are forming (Farmer et al., 1986) (Hordijk et al., 2010) (Kauffman, 1986); in this work we present a first attempt toward a dynamical detection of these systems.

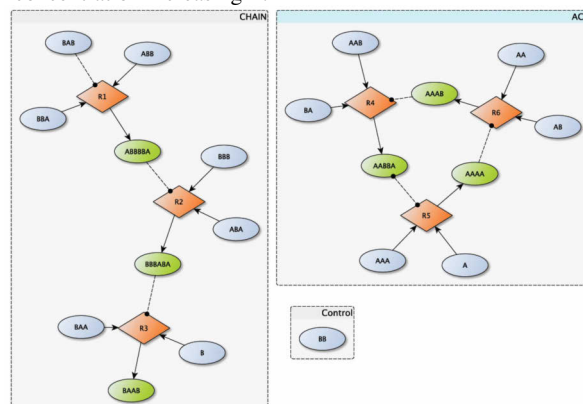
We use a simple system (inspired by a model (Filisetti et al. 2011a) (Filisetti et al. 2011b) (Filisetti et al. 2011c) (Farmer et al., 1986) originally due to Kauffmann (Kauffmann, 1993) (Kauffmann, 1995)) where there are two distinct reaction pathways, a linear reactions chain (CHAIN) and an autocatalytic set of molecular species (ACS) (see figure 2): both reactions pathways occur in an open well-stirred chemostat (CSTR) with a constant influx of feed molecules and a continuous outgoing flux of all the molecular species proportional to their concentration. The dynamics of the system is described adopting a deterministic approach whereby the reaction scheme is translated in a set of Ordinary Differential Equations (ODE) integrated by means a fourth-order Runge-Kutta method (Young and Gregory, 1988).

The main entities of the model are molecular species ("polymers") represented by linear strings of letters A and B, forming together a catalytic reactions system composed of 6 distinct condensation reactions in which two species are glued to create a longer species. The reactions occur only in presence of a specific catalyst, since spontaneous reactions are assumed to occur too slowly to affect the system behavior. Accordingly, in the following the reaction scheme is presented:



According to the three molecular nature of the condensation reaction, reactions occur in 2 two steps: in the former the catalyst binds the first substrate forming a molecular complex, while in latter the molecular complex binds the second substrate releasing the product and the catalyst. The "food set" of the linear chain ( $BAB \rightarrow ABBBA \rightarrow BBBABA \rightarrow BAAB$ ) is formed by the species  $ABB$ ,  $BBA$ ,  $BBB$ ,  $ABA$ ,  $BAA$ ,  $B$ , whereas the food set of and the autocatalytic cycle ( $AABBA \rightarrow AAAA \rightarrow AAAB \rightarrow AABBA$ ) is formed by the species  $BA$ ,  $AAB$ ,  $AAA$ ,  $A$ ,  $AB$ ,  $AA$ . Besides, an independent molecular species  $BB$  not involved in any reactions has been introduced as control species (figure 2).

The asymptotic behavior of this kind of systems is a single fixed point (Vasas et al., 2012), due to the system feedback structure. In order to apply our analysis we need to observe the feedbacks in action, therefore we perturb the concentration of some molecules in order to trigger a response in the concentration of (some) other species. So we temporarily set to zero the concentration of some species (in the example of fig.2 of the species  $ABBBA$ ,  $BBBABA$ ,  $AABBA$ ,  $AAAA$ ,  $AAAB$ ) after the system reached its stationary state<sup>2</sup>: in order to analyze the system response to perturbations we use a 3-level coding, where for each species the digit '0'-'1'-'2' stand respectively for "concentration decreasing", "no change" and "concentration increasing"<sup>3</sup>.

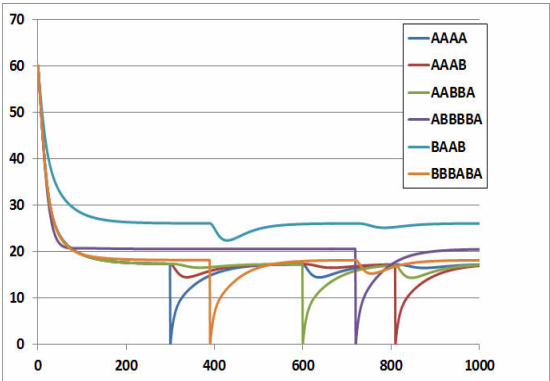


**Figure 2** The chemical system under analysis. Circular nodes depict chemical species, the blue ones stand for those injected on the CSTR (food species) and the green ones represent the more complex species built by specific concatenations of the food species, see reaction scheme in the text. Diamond shapes represent reactions where incoming arrows go from substrates to reactions and outgoing arrows go from reactions to products. Dashed lines indicate the catalytic role of a particular molecular species within the specific reaction context. The kinetic constants of all present reactions have the same value  $k_{dir}=0.0025 \text{ s}^{-1} \text{ mol}^{-1}$ ; the incoming concentration of each food species is 1.0 mol, whereas each second the 2% of the CSTR volume is renewed

<sup>2</sup> In this example the analyzed data series starts from second 200, in order to avoid the initial transient

<sup>3</sup> In such a way we can abstract from the different concentration present on the system, a species being constant if its concentration change from previous time instant is below the threshold of 0.1%

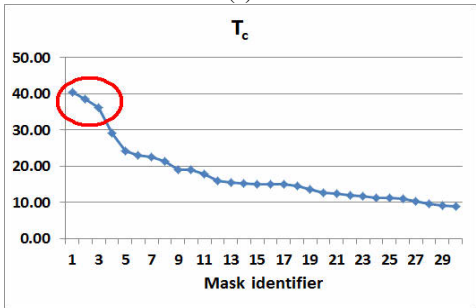
The results clearly indicate the presence of two distinct systems of size 3 (the second and third rows in fig.4a) that correspond to CHAIN and ACS. Note that the leave of CHAIN (BAAB) is not strongly affected by the zeroing of ABBBA species (because the perturbation of this species, root of the linear chain, affects only in a limited manner the following species BBBABA, whose change in turn even lesser affects the concentration of species BAAB...): this attenuation process induces a dynamical hierarchy on CHAIN system, which allows the finer subdivision highlighted by the first row of fig.4a. This phenomenon is absent on ACS, a more homogeneous system where no roots are present.



**Figure 3** The chemical system trajectory, including the performed perturbations (only products are analyzed)

Cluster size	AAAA	AAAB	AABBA	ABBBBA	BAAB	BBBABA	Tci
2							40.34
3							38.42
3							36.19
4							29.10
3							24.10
2							22.98
4							22.58
2							21.41
3							18.93
5							18.92

(a)



(b)

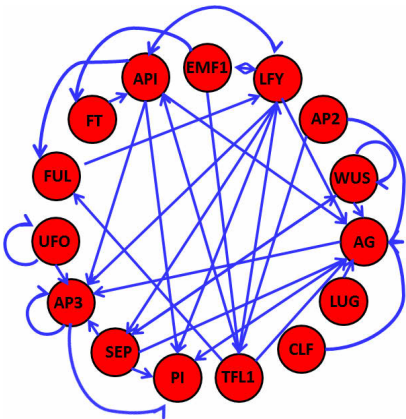
**Figure 4** (a) The masks resulting from the chemical system analysis and (b) their corresponding Tc values. Note that the three masks whose Tc values outperform the other ones correctly identify the system's components (see text for details)

### Arabidopsis thaliana

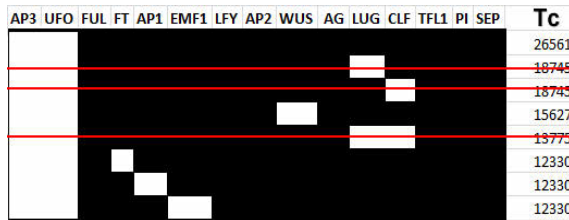
It is possible to expand the analysis to BN derived from biological data of specific living beings. In this work we take advantage from the available data of the gene regulatory network shaping the developmental process of Arabidopsis thaliana: although the whole network is largely unknown, a certain subsystem has been identified as responsible for the floral organ specification. We will not enter here a discussion about the merits and limits of this simplified model, but we will take it "for granted" and we will apply our method to test whether it can discover significant MDSs.

The network is modeled by means of a BN described in (Chaos et al., 2006), which has 15 nodes and 10 different attractors (all fixed points): we therefore build a data series containing a number of repetitions of these attractors in proportion to their basins of attraction. In doing so it is possible to note that genes LUG and CLF are constantly active in all the attractors: this particular feature introduces a particular "noise" on CI analysis, by adding spurious cluster among the first positions. Indeed, it is possible to analytically demonstrate that the addition of constant nodes in clusters with high  $T_c$  leads again to other clusters with high  $T_c$  values: these additions nevertheless do not have particular biological meanings (the added elements do not introduce any variation), so the corresponding clusters can be memorized as "not significant".

The analysis clearly groups genes UFO and AP3, present alone on the best significant cluster and in all the following 20 most significant clusters. Note that the second significant cluster includes gene WUS: indeed, for biologists (Lenhard et al., 2001) (Lohman and others 2001) UFO and WUS are key inputs for determining the specific time and site where the combinations of gene activities considered in the developmental process are established, whereas AP3 is an important transcription factor. So, our analysis perceives the combination of a "sensor" (UFO) and of an influential "signaler" (AP3) as a single powerful dynamical engine, whose action can be tuned by WUS gene, demonstrating that it could highlight biologically interesting functional relationships.



**Figure 5** Floral network of *A. thaliana* (from (Chaos et al., 2006))



**Figure 6** Matrix illustrating the elements of the clusters (white in figure) identified by our analysis and their corresponding  $T_c$  values. Genes LUG and CLF are always constant along all the attractors and therefore their insertion in “active” MDSs can be excluded a priori (it is possible to analytically demonstrate that the addition of constant nodes in already existing clusters leads to cluster with high  $T_c$  values – but these additions do not seem have particular biological meanings)

## Conclusions

In this paper we introduced a system to identify candidate emergent mesolevel dynamical structures in dynamical networks. The main novelty of the present work, in comparison to previous application of the cluster index and of similar measures (Tononi et al. 1998) is that we used it to consider truly dynamical networks, and not only fluctuations around stable asymptotic states.

Future works will consider the application of the method to other important natural and artificial networks, an improved understanding of its working and the use of entropies taken at different times.

As examples of application we used time series of simple artificial systems and more complex data coming from catalytic reaction networks and biological gene regulatory systems (*A.thaliana*). The analysis performed by our system was able to identify correctly the MDSs, and we think it could suggest interesting new ways in dealing with artificial and biological systems.

Future work will consider the application of the method to other important natural and artificial networks, with the aim of deepen our understanding of its working principles and assessing its analysis power. In addition, we also plan to extend the definition of cluster index so as to take into account time relationships, for example by using of entropies taken at different times.

**Acknowledgments.** This article has been partially funded by the UE projects “MD – Emergence by Design”, Pr.ref. 284625 and “INSITE - The Innovation Society, Sustainability, and ICT” Pr.ref. 271574, under the 7th FWP - FET programme.

## References

1. Benedettini S. Identifying mesolevel dynamical structures ECLT (European Center for Living Technologies) technical report, Venice 2013
2. Carletti, T., Serra, R., Poli, I., Villani, M. & Filisetti, A. (2008): Sufficient conditions for emergent synchronization in protocell models. *Journal of Theoretical Biology* - Elsevier Ltd. v. 254 pp.741–751
3. Chaos A., Aldana m., Espinosa-Soto C., Ponce de Leon B.G., Garay Arroyo A., Alvarez-Buylla E.R. (2006) From Genes to

- Flower Patterns and Evolution: Dynamic Models of Gene Regulatory Networks *J Plant Growth Regul* 25:278–289
4. Emmeche, C., Köppe, S., Stjernfelt, F. (1997) Explaining emergence: Towards an ontology of levels. *Journal for General Philosophy of Science*, 28:83–119
5. Farmer, J., Kauffman, S.A., Packard N. (1986) Autocatalytic replication of polymers. *Physica D: Nonlinear Phenomena*. 220, 50–67
6. Filisetti A., Serra R., Carletti T., Villani M., Poli i. (2010) Non-linear protocell models: Synchronization and Chaos *Eur. Phys. J. B* 77, 249–256
7. Filisetti A., Graudenzi A., Serra R., Villani M., De Lucrezia D., Fuchslin R.M., Kauffman S.A., Packard N., Poli I. (2011a) A stochastic model of the emergence of autocatalytic cycles *Journal of Systems Chemistry*, 2:2 doi:10.1186/1759-2208-2-2
8. Filisetti A., Graudenzi A., Serra R., Villani M., Fuchslin R.M., Packard n., Kauffman S.A., Poli I (2011b) A stochastic model of autocatalytic reaction networks *Theory in Biosciences* v.130 Springer Berlin/Heidelberg DOI 10.1007/s12064-011-0136-x
9. Filisetti, A., Graudenzi, A., Serra, R., Villani, M., De Lucrezia, D., and Poli, I. (2011c). The role of energy in a stochastic model of the emergence of autocatalytic sets. In *Advances in Artificial Life ECAL 2011 Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*, H. Bersini, P. Bourguin, M. Dorigo, and R. Doursat, eds. (MIT Press, Cambridge, MA), pp. 227–234.
10. Ganti, T.: *Chemoton Theory, Vol. I: Theory of fluid machineries: Vol. II: Theory of living system*. New York: Kluwer Academic/Plenum. (2003)
11. Haken H. (2004) *Synergetics* Springer-Verlag Berlin Heidelberg
12. Hordijk, W., Hein J., Steel m. (2010) Autocatalytic Sets and the Origin of Life. *Entropy*. 12, 7, 1733–1742
13. Kauffman, S.A. (1986) Autocatalytic sets of proteins. *J Theor Biol*. 119, 1, 1–24
14. Kauffman, S.A., 1993. *The Origins of Order*. Oxford University Press, Oxford.
15. Kauffman, S.A., 1995. *At Home in the Universe*. Oxford University Press, Oxford.
16. Lane, D., van der Leeuw, S., Pumain, D., West, G. eds. (2009) Complexity perspectives on innovation and social change. Springer-Verlag Berlin Heidelberg.
17. Lenhard M, Bonhart A, Jurgens G, Laux T. (2001) Termination of stem cell maintenance in Arabidopsis floral meristems by interactions between WUSCHEL and AGAMOUS. *Cell* 105:805–808
18. Lohmann JU, Hong RL, Hobe M, Busch MA, Parcy F, others. (2001) A molecular link between stem cell regulation and floral patterning in Arabidopsis. *Cell* 105:793–803.
19. Luisi, P.L., Ferri F., Stano P. (2006) Approaches to semi-synthetic minimal cells: a review. *Naturwissenschaften*. 93, 1, 1–13
20. Mansy, S.S., Schrum J.P., Krishnamurthy M., Tobé S., Treco I D.A., Szostak J.W. (2008) Template-directed synthesis of a genetic polymer in a model protocell. *Nature*. 454, 7200, 122–125
21. Rasmussen, S. Chen L, Nilsson M, Abe S. (2003) Bridging nonliving and living matter. *Artificial life*. 9, 3, 269–316
22. Salthe, S.N. (1985) *Evolving Hierarchical Systems: Their Structure and Representation*. Columbia University Press.
23. Serra, R., Villani, M. & Semeria A. (2004): Genetic network models and statistical properties of gene expression data in knock-out experiments *Journal of Theoretical Biology* 227: 149–157
24. Serra R., Villani M., Barbieri B., Kauffman S.A., Colacci A. (2010) On the dynamics of random boolean networks subject to noise: attractors, ergodic sets and cell types *Journal of Theoretical Biology* - Elsevier Ltd. v.265 pp.185–193

25. Shmulevich, I., Kauffman, S.A., Aldana, M., 2005. Eukaryotic cells are dynamically ordered or critical but not chaotic. *Proc. Natl Acad. Sci.* 102, 13439–13444.
26. Solé, R.V., Munteanu A, Rodríguez-Caso C, Macía J. (2007) Synthetic protocell biology: from reproduction to computation. *Philos Trans R Soc Lond B Biol Sci.* 362, 1486, 1727–1739 (2007)
27. Stano, P., Luisi, P.L. (2010) Achievements and open questions in the selfreproduction of vesicles and synthetic minimal cells. *Chemical communications* (Cambridge, England). 46, 21, 3639–53
28. Szostak, J.W. Bartel D.P., Luisi P.L. (2001) Synthesizing life. *Nature.* 409, 6818, 387–390
29. Tononi G., McIntosh A.R., Russell D.P., Edelman G.M. (1998) Functional Clustering: Identifying Strongly Interactive Brain Regions in Neuroimaging Data *NEUROIMAGE* 7, 133–149
30. Vasas V., Fernando C., Santos M., Kauffman S.A., Szathmáry E. (2012) Evolution before genes *Biology Direct*, 7:1
31. Villani M., Serra, R., Graudenzi, A. & Kauffman, S.A. (2007): Why a simple model of genetic regulatory networks describes the distribution of avalanches in gene expression data. *J. Theor. Biol.* 249 : 449–460
32. Villani M, Barbieri A, Serra R (2011) A Dynamical Model of Genetic Networks for Cell Differentiation. *PLoS ONE* 6(3): e17703. doi:10.1371/journal.pone.0017703
33. Villani M., Serra R. (2013) On the dynamical properties of a model of cell differentiation *EURASIP Journal on Bioinformatics and Systems Biology*, 2013:4 – Springer
34. Villani M., Benedettini S., Roli A., Lane D., Poli I., Serra R. (2013) Identifying emergent dynamical structures in network models *Proceeding of WIRN 2013*, submitted
35. Young D.M., Gregory R.T. (1988) A Survey of Numerical Mathematics *Dover Publications, Incorporated*



# On the evolution of self-organised role-allocation and role-switching behaviour in swarm robotics: a case study

Elio Tuci<sup>1</sup>, Boris Mitavskiy<sup>2</sup> and Gianpiero Francesca<sup>3</sup>

<sup>1,2</sup>Aberystwyth University, Llandinam Building, Aberystwyth, SY23 1RE, UK

<sup>3</sup>IRIDIA-ULB, Av. Franklin Roosevelt 50, CP 195/6, 1050 Brussels, Belgium  
elt7@aber.ac.uk<sup>1</sup>, bom4@aber.ac.uk<sup>2</sup>, gfrancesca@iridia.ulb.ac.be<sup>3</sup>

## Abstract

In spite of its significance for the adaptability of autonomous robotic swarms, the dynamic allocation and re-distribution of robots to tasks (i.e., role-allocation and role-switching behaviour) is still a design challenge in swarm robotics. This study investigates a simulated foraging scenario in which the variability of the environmental conditions requires that robots switch between two roles (i.e., foraging and nest-patrolling). To the best of our knowledge, this is the first simulation study that demonstrates that role-allocation and role-switching behaviour can be evolved using dynamic neural network controllers for robots with minimal communication capabilities. Initial analyses of the best evolved teams shed light on some of the characteristics and robustness of the strategies used by these teams to repeatedly face this task.

## Introduction

Swarm robotics is a particular approach to the design of multi-robot systems that finds its theoretical roots in recent studies in animal societies, such as ants and bees (Dorigo and Şahin, 2004). Despite noise in the environment, errors in processing information and performing tasks, and no global information, social insects are quite successful at performing group-level tasks (Anderson et al., 2001; Camazine et al., 2001). Based on the social insect metaphor, swarm robotics emphasises aspects such as decentralisation of the control, limited communication abilities among robots, use of local information, emergence of global behaviour and robustness. These properties are meant to facilitate the design of artificial systems scalable to group size, robust to noise, and adaptive to environmental changes.

Research in swarm robotics has been focusing on mechanisms to enhance the efficiency of the group through some form of cooperation among the individual agents. Complex forms of group cooperative responses can require task-partitioning (i.e., division of a collective task into individual sub-tasks) and/or task/role-allocation (i.e., allocation of sub-tasks/roles to different individuals, see Labella et al., 2006). The latter can be a dynamic and flexible process, in that the number of individuals engaged in any given task may need to continually change, as circumstances require. Hereafter,

we use the term role-switching or task-switching behaviour to refer to the process in which one or more agents leave their current activity to join a different one for the benefit of the team. In spite of its significance for the adaptability of the swarm, the autonomous and dynamic re-distribution of robots to tasks is still a design challenge. This study aims to investigate, in a simulated scenario, the conditions for the emergence of dynamic role-allocation and role-switching behaviour in teams of autonomous agents.

We face this challenge using the Evolutionary Robotics (ER) design method. ER is based on a bottom-up *modus operandi*, where variations are introduced at the genetic level, and selection is performed on the basis of the effects that genetic variations have on the global behaviour of the swarm. With respect to other design methods, ER does not require the designer to make strong assumptions concerning what behavioural and communication mechanisms are needed by the robots. Individual behavioural strategies and rules of actions are determined by the evolutionary process that favours (through selection) those solutions which improve an agent's and the group's ability to accomplish the collective task. The operational mechanisms of the best evolved teams can be *a posteriori* analysed to gain insight into the solutions of the collective problem.

Our long term goal is to apply ER to learning, i) the evolutionary dynamics underpinning the emergence of the role-allocation and role-switching behaviour in autonomous robots; ii) the nature of the individual mechanisms underlying the group response. This study is the first step in this direction. We investigate a scenario in which teams of homogeneous robots are required to split in foragers and nest-patrollers (i.e., robots that remain in the nest). The task demands that both roles are played, but according to different rules. Under one condition, foragers have to be more numerous than patrollers, while under the other the situation is the opposite. The results show that, such a relatively complex team behaviour can be obtained using only very limited means of interaction (i.e., infra-red sensors). Initial analyses of the best evolved teams shed light on some of the characteristics and robustness of the behavioural strategies used by

these teams to repeatedly face this task.

## Background and motivations

Several studies in swarm robotics are focused on issues related to the conditions that facilitate the emergence of behavioural specialisation among the robots. Many of these studies provide a comparative cost/benefit analysis of engineering specialisation by using heterogeneous teams (i.e., teams of robots with different individual controllers) versus dynamic and emergent forms of specialisation in homogeneous teams (i.e., team of robots that share the same individual controller, see Eiben et al., 2007; Luke and L.Spector, 1996; Bongard, 2000; Ijspeert et al., 2001; Quinn, 2001; Tuci and Trianni, 2012). From an evolutionary design perspective, homogeneous groups tend to imply a smaller search space than heterogeneous groups, and contrary to the latter they are not affected by the credit-assignment problem (i.e., the problem of divvying up among the team members the reward received through their joint actions, (see Panait and Luke, 2005)). Nevertheless, in those contexts in which partitioning the group task into different sub-tasks it is beneficial for the team, single controllers in homogeneous groups have to underpin behavioural skills required by the robots to undertake all the sub-tasks, as well as the decision making related to the allocation of robots to sub-tasks. The studies described in (Quinn et al., 2003; Ampatzis et al., 2009) showed that, in relatively small groups (2 to 3 robots), task-allocation and specialisation of the team members can be developed by breaking the homogeneity condition through the evolution of ritualised coupled behaviours subject to the effect of random noise inherent in sensory and motor hardware components. However, these studies looked at scenarios in which, once specialised, the agents do not need to reconsider their roles within the group. We know that in natural swarms task-allocation is quite dynamic and flexible, in that the number of individuals engaged in any given task continually changes as circumstances require (Gordon, 1996). Empirical evidence shows that internal factors, such as genetic and morphological differences among the workers, do not always account for the individual variability in task preferences. Single workers in various species of ants perform a variety of tasks, changing from one task to another by tracking contingent factors such as environmental stimuli, the number of agents currently engaged in other tasks, or the rate of interactions with other agents. For example, when a new food source suddenly become available to an harvester ant colony, which competes with other seed-eating species for food, ants previously engaged in other tasks will switch to foraging (Davidson, 1977).

Biologists are particularly interested in models focused on the evolution of emergent principles underpinning task-allocation. As stated in (Duarte et al., 2011), "... disappointingly few attempts have been made to develop realistic scenarios for how the mechanism underlying self-organised

division of labour evolve over the course of generation...". This is partially due to the limitations of classic methodologies at disposal of biologists. The authors in (Duarte et al., 2011) state that evolutionary models of division of labour tend to focus on the conditions in which specialisation is better than generalist strategies, ignoring the mechanism through which specialisation may arise. On the other hand, self-organisation models do not consider the evolutionary trajectories that may lead to task-allocation. Our long-term aim is to create models of the evolution of self-organised role-allocation and role-switching behaviour that can be used as "intuition-pump" for indicating potential evolutionary drivers and emergent behavioural rules capable of accounting for the collective behaviour of natural swarms (see Vassie and Morlino, 2012, for an epistemological account of robotic models). For example, biological evidence shows that behavioural specialisation in various insect societies evolve to minimise the costs of task-switching. Nevertheless, the fact that individuals switch tasks indicates that the evolution of task-allocation systems can not be merely the production of genetically and/or morphologically different individuals, each suited to a particular tasks. Empirically, little is still known about the selective pressures for task-switching behaviour. Robotics models can represent effective alternative methodological tools to investigate these issues. Our aim is to recreate these types of self-organised dynamics in homogeneous robotic swarms by evolving the individual mechanisms and rules of interactions/communication underpinning task-allocation and task-switching behaviour.

## The Simulation Environment

In the foraging scenario studied in this paper, the environment is a boundless arena with a nest and a foraging site. The nest is a circular area indicated by a green light, in which the colour of the floor is in shades of grey. The foraging site is also a circular area indicated by a red light, in which the colour of the floor is in a different shades of grey with respect to the nest. The colour of the arena floor is white. The radius of both the nest and the foraging site is randomly defined at the beginning of each trial in the interval [20cm, 30cm]. Both lights, the green one located in the nest and the red one located in the foraging site, are positioned 6cm above the floor and, when turned on, they are visible from everywhere within the arena. In each trial, the green light is placed at the centre of the nest. The red light is randomly placed anywhere within a semicircular area of 10cm radius centred in the centre of the foraging site. The centre of the nest is 1m far from the centre of the foraging site (see Fig. 1a)

The robots kinematics are simulated using a modified version of the "minimal simulation" technique described in (Jakobi, 1997). Our simulation models a e-puck robot, a 3.55cm radius cylindrical robot. It is provided with eight

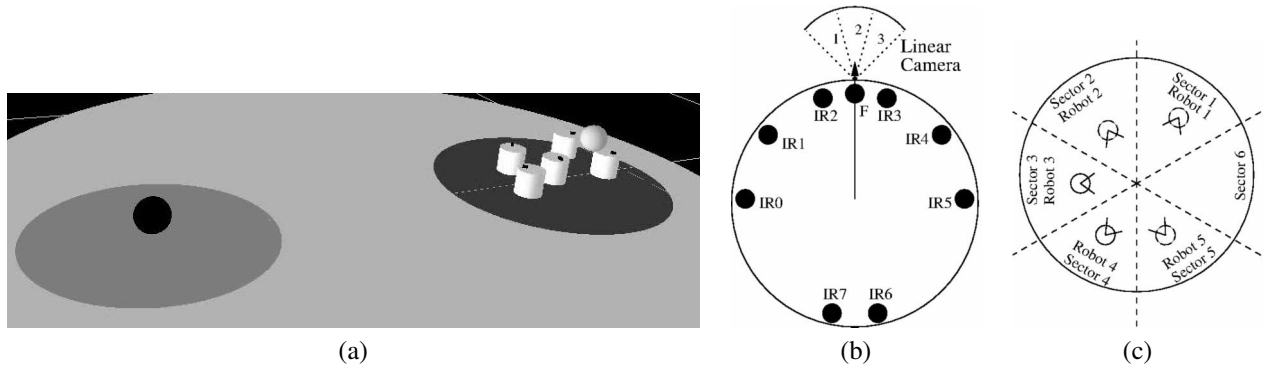


Figure 1: (a) Experimental scenario (snapshot taken in *Env. A*, during *Phase 1*) showing the nest (dark grey circle), the foraging site (light grey circle), the five robots (white cylinders in the nest), and the lights (small circles above the nest and foraging site). (b) E-puck body-plan. The black circles refer to the position of the infra-red (IR), and floor sensor ( $F$ ) positioned facing downward on the underside of the robot (see Fig. 1b). The dotted lines indicate the robot's view with the three camera's sectors. (c) Robots starting position. The big circle represents the nest. The small circles are the robots, and the short black segments indicate the angle within which the robot orientation is chosen.

infra-red sensors ( $IR^i$  with  $i = \{0, \dots, 7\}$ ), which give the robot a noisy and non-linear indication of the proximity of an obstacle (in this task, an obstacle can only be another robot); a linear camera to see the lights; and a floor sensor ( $F$ ) positioned facing downward on the underside of the robot (see Fig. 1b). The IR sensor values are extrapolated from look-up tables provided with the Evorobot\* simulator (see Nolfi and Gigliotta, 2010). The  $F$  sensor can be conceived of as an IR sensor capable of detecting the intensity of grey of the floor. It returns 0 if the robot is on white floor, 0.5 if it is on light grey floor, and 1 if it is on dark grey floor. The robots camera has a receptive field of  $30^\circ$ , divided in three equal sectors, each of which has three binary sensors ( $C_i^B$  for blue,  $C_i^G$  for green, and  $C_i^R$  for red, with  $i = \{1, 2, 3\}$  indicating the sector). Each sensor returns a value which is 0 if no light is detected, 1 when a light is detected. The camera can detect coloured objects up to a distance of 1.5m. The robots can not see each other through the camera. The robot has left and right motors which can be independently driven forward or reverse, allowing it to turn fully in any direction. The robot maximum speed is 8cm/s.

### The Task and the Fitness Function

Teams comprising five simulated e-puck robots are evaluated in the context of a dynamic role allocation and role switching behaviour. By taking inspiration from the behaviour of social insects, the roles are nest patrolling and foraging (hereafter, we refer to them as *role P*, and *role F*, respectively). Roughly speaking, *role P* requires a robot to remain within the nest. *Role F* requires a robot to leave the nest for the foraging site, to spend a certain amount of time at the foraging site, and then to come back to the nest. A team is required to execute both roles simultaneously. Therefore, the robots have to go through a role-allocation phase in which they autonomously decide who is doing what, and then ex-

ecute their respective roles.

Moreover, the robots are required to be able to switch from one role to the other (i.e., role switching behaviour) due to the fact that they experience two different types of environment, *Env. A* and *Env. B*. In *Env. A*, *role F* is more important than *role P*. This means that in *Env. A*, a team maximises the fitness if the majority of robots (i.e., more than two robots) visits the foraging site and the minority (i.e., less than three robots) remains in the nest. In *Env. B*, *role P* is more important than *role F*. This means that a team maximises the fitness if the majority of robots (i.e., more than two robots) remains in the nest and the minority (i.e., less than three robots) visits the foraging site. Since a team, throughout its life-span, experiences twice both types of environment, not all the robots can specialise on a single role. The robots have to be able to play both roles and eventually to switch from one role to the other based on the current environmental condition and the roles allocated to the other team mates. How can a robot distinguish between *Env. A* and *Env. B*? The two types of environment can be distinguished by the intensity of grey colouring the floor in the nest site. In *Env. A*, the nest is coloured in dark grey and the foraging site in bright grey. In *Env. B*, the nest is coloured in bright grey and the foraging site in dark grey.

During evolution, each team undergoes a set of  $E = 2$  evaluation sequences (hereafter, e-sequence). An e-sequence is made of  $V = 4$  trials, in which the teams experience twice each type of environment in the following order: trial 1 *Env. A*, trial 2 *Env. B*, trial 3 *Env. A*, trial 4 *Env. B*. At the beginning of trial 1 of each e-sequence, the robots controllers are reset, and each robot is randomly placed within an area corresponding to a sector of the nest. The nest is divided in 6 sectors, and the robots are placed in sector 1 to 5, as illustrated in Fig. 1c. Each robot is randomly oriented in a way that the light can be within an angular distance of

$\pm 36^\circ$  from its facing direction (see Fig. 1c).

Each trial differs from the others in the initialisation of the random number generator, which influences the robots initial position and orientation, all the randomly defined features of the environment, and the noise added to motors and sensors (see Jakobi, 1997, for further details on sensors and motor noise). Within a trial, the team life-span is  $T=900$  simulation cycles (with 1 simulation cycle lasting 0.1s). Robots are frozen (i.e., don't move and do not contribute to the team fitness) if they exceed the arena limits (i.e., a circle of 120cm radius, centred in the middle point between the nest and the foraging site). Trials are terminated earlier if all the robots are frozen, or the team exceeds the maximum number of collisions (i.e., 10). In trials following the first one of each e-sequence (trial 2,3, and 4), the robots are repositioned only if the previous trial has been terminated earlier, or with one or more robot frozen.

Each trial is divided into three phases. During *Phase 1*, which lasts 12s, the green light is on and the red light is off. The robots are required to stay within the nest. During *Phase 2*, which can last from a minimum of 47,5s to a maximum of 52.5s, the red light is on and the green light is off. During *Phase 2*, a team is required to behave according to the rules of the task. That is, in *Env. A*, the majority of robots (i.e., more than two robots) has to visit the foraging site and the a minority (i.e., less than three robots) has to remain for the entire length of this phase in the nest. In *Env. B*, the majority of robots has to remain for the entire length of *Phase 2* in the nest and the minority has to visit the foraging site. A robot is considered having visited the foraging site if, during *Phase 2*, it spends more then 100 consecutive time steps within the foraging site. During *Phase 3*, which starts at the end of *Phase 2* and terminates at the end of the trial, the green light is on again and the red light is off. The robots that were foraging during *Phase 2* are required to return in the nest to rejoin their team mates.

The fitness of a genotype is its average team evaluation score after it has been assessed for two e-sequences (i.e., for a total of 8 trials). In each trial ( $v$ ) of each e-sequence ( $e$ ), the team is rewarded by an evaluation function  $F_{ev}$  which corresponds to:  $F_{ev} = ((PH^1 \times PH^3) + PH^2) \times PEN$ .  $PH^1 \in [0, 1]$  is computed during *Phase 1*, and it corresponds to the robot average proportion of time steps “inside” the nest.  $PH^2$  is computed during *Phase 2*.  $PH^2 = 5$  if the robots follow the rules of the task (see above). If the team does not behave according to the rules of the task, then  $PH^2$  corresponds to i) the proportion of foraging robots multiplied by two, in *Env. A* ii) the proportion of robots remained in the nest multiplied by two in *Env. B*.  $PH^3$  is computed during *Phase 3*.  $PH^3 = 2$  if all the robots terminate the phase within the nest.  $PH^3$  corresponds to the proportion of robots that terminated the trail within the nest, otherwise.  $PH^3$  is set to 1 if the trial is terminated before a team reaches *Phase 3*. The team collision penalty

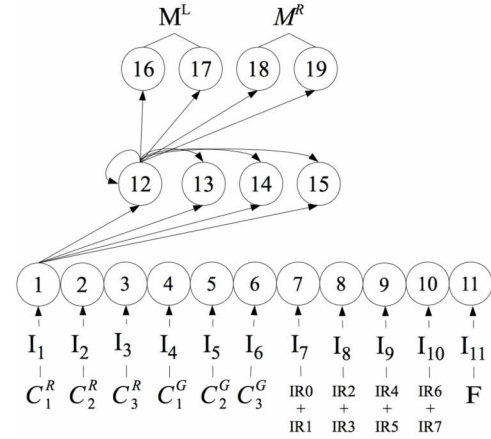


Figure 2: The neural network.

$PEN$  is inversely proportional to the number of collisions, with  $PEN = 1$  with no collisions, and  $P^a = 0.4$  with 10 collisions in a trial; The average team evaluation score is  $F = \frac{1}{E} \frac{1}{V} \sum_{e=1}^E \sum_{v=1}^V F_{ev}$ .

### Controller and the Evolutionary Algorithm

The robot controller is composed of a continuous time recurrent neural network (CTRNN) of 11 sensor neurons, 4 inter-neurons, and 4 motor neurons (see Beer and Gallagher, 1992). The structure of the network is shown in Fig. 2. The states of the motor neurons are used to control the speed of the left and right wheels as explained later. The values of sensory, internal, and motor neurons are updated using equations 1, 2, and 3.

$$y_i = gI_i; \text{ for } i \in \{1, \dots, 11\}; \quad (1)$$

$$\tau_i \dot{y}_i = -y_i + \sum_{j=1}^{15} \omega_{ji} \sigma(y_j + \beta_j); \text{ for } i = \{12, \dots, 15\}; \quad (2)$$

$$y_i = \sum_{j=12}^{15} \omega_{ji} \sigma(y_j + \beta_j); \text{ for } i = \{16, \dots, 19\}; \quad (3)$$

with  $\sigma(x) = (1 + e^{-x})^{-1}$ . In these equations, using terms derived from an analogy with real neurons,  $y_i$  represents the cell potential,  $\tau_i$  the decay constant,  $g$  is a gain factor,  $I_i$  with  $i = \{1, \dots, 11\}$  is the activation of the  $i^{th}$  sensor neuron (see Fig. 2 for the correspondence between robot's sensors and sensor neuron),  $\omega_{ji}$  the strength of the synaptic connection from neuron  $j$  to neuron  $i$ ,  $\beta_j$  the bias term,  $\sigma(y_j + \beta_j)$  the firing rate (hereafter,  $f_i$ ). All sensory neurons share the same bias ( $\beta^I$ ), and the same holds for all motor neurons ( $\beta^O$ ).  $\tau_i$  and  $\beta_i$  with  $i = \{12, \dots, 15\}$ ,  $\beta^I$ ,  $\beta^O$ , all the network connection weights  $\omega_{ij}$ , and  $g$  are genetically specified networks' parameters. At each time step, the output of the left motor is  $M^L = f_{16} - f_{17}$ , and the right motor is  $M^R = f_{18} - f_{19}$ , with  $M_L, M_R \in [-1, 1]$ . Cell potentials are set to 0 when



the network is initialised or reset, and equation 2 is integrated using the forward Euler method with an integration time step  $\Delta T = 0.1$ .

A simple evolutionary algorithm using linear ranking is employed to set the parameters of the networks (Goldberg, 1989). The population contains 100 genotypes. Generations following the first one are produced by a combination of selection with elitism, recombination, and mutation. For each new generation, the three highest scoring individuals (“the elite”) from the previous generation are retained unchanged. The remainder of the new population is generated by fitness-proportional selection from the 60 best individuals of the old population. Each genotype is a vector comprising 87 real values (76 connections, 4 decay constants, 6 bias terms, and a gain factor). Initially, a random population of vectors is generated by initialising each component of each genotype to values chosen uniformly random from the range  $[0,1]$ . New genotypes, except “the elite”, are produced by applying recombination and mutation. Each new genotype has a 0.3 probability of being created by combining the genetic material of two parents. During recombination, one crossover point is selected. Genes from the beginning of the genotype to the crossover point is copied from one parent, the other genes are copied from the second parent. Mutation entails that a random Gaussian offset is applied to each real-valued vector component encoded in the genotype, with a probability of 0.04. The mean of the Gaussian is 0, and its standard deviation is 0.1. During evolution, all vector component values are constrained to remain within the range  $[0,1]$ .

## Results

Our objective is to design neuro-controllers for homogeneous teams of robots required to exhibit the following skills: dynamic role-allocation, and role-switching behaviour. This means that teams have to be capable of dynamically allocating roles to robots and simultaneously executing both roles in each trial. Moreover, not all the robots can specialise in a single role (i.e., playing only a single role throughout an e-sequence). This is because there are two different types of environment: *Env. A*, in which the majority of the robots has to play *role F*; and *Env. B*, in which the majority of the robots has to play *role P*. The robots (at least one) have to switch role between consecutive trials for the majority to be distributed as required by the task.

10 evolutionary runs, each using a different random initialisation were carried out for 2500 generations. Seven evolutionary runs managed to generate teams with the highest fitness score (data not shown<sup>1</sup>). In order to have a better estimate of the behavioural capabilities of the evolved teams, we post-evaluated, for each run, the fittest team of each generation for the last 500 generations. The post-evaluation test

<sup>1</sup>See <http://users.aber.ac.uk/elt7/suppPagn/ECAL2013/suppMat.html> for further methodological details, graphs and movies of the best teams.

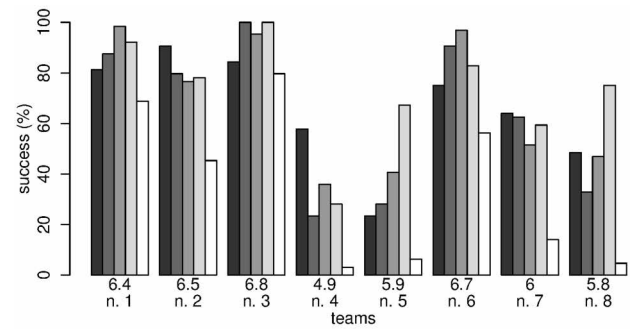


Figure 3: Graph showing the performances of the best teams from 8 different evolutionary runs, re-evaluated for 64 sequences. For each team, the number below the bars refers to the average score  $F$ . The four bars in shades of grey (from black to light grey) indicate the % of success in each of the four temporal sections (or trials) of an e-sequence, respectively. The fifth bar (the white one) indicates the % of successful e-sequences out of 64.

consists of 64 e-sequences per team (for a total of 256 trials, 4 trials times 64 e-sequences). The performance of each team is measured using the metrics  $F$  illustrated above, with  $E = 64$ . For each run, the team with the highest average re-evaluation score  $F$  is assumed to be an adequate measure of the success of the run.

The graph in Fig. 3 shows for each best team, the fitness ( $F$ ), the success rate (%) in each temporal section (i.e., trial) of the e-sequence (i.e., black bars for the first trials, dark grey bars for the second trials, medium grey bars for the third trials, and light grey bars for the fourth trials), and the percentage of successful e-sequence out of 64 (see white bars). An e-sequence ( $e$ ) is considered successful if a team manages to get the highest score ( $F_{ev} = 7$ ) in each of the four trials ( $v$ ). Note that, the results of the worst two runs have been omitted from Fig. 3 because the percentage of successful e-sequence of the best teams was 0.

The numbers just below the bars in Fig. 3 refers to the average fitness score of each team. Four teams (team n. 1, 2, 3, and 6) managed to get an average fitness score quite close to the optimum  $F = 7$ , with the team generated by run n. 3 (hereafter, team n. 3) being the most successful. The bars in the graph explain the meaning of these fitness scores in term of teams’ performance and robustness. For example, team n. 1, 2, 3, and 6, have a relatively high success rate in each of the four temporal sections of the e-sequence (see black to light grey bars in Fig. 3, for team n. 1, 2, 3, 6). Nevertheless, these scores do not necessarily correspond to a high percentage of successful e-sequences. Team n. 1, 2, 6 have a percentage of successful e-sequence below 70% (see white bars in Fig. 3, for team 1, 2, and 6). Team n. 3, instead, manages to successfully complete about 80% of the

64 post-evaluation e-sequences, proving to be quite robust and effective. As shown in Table 1, team 3 is also the only team for which unsuccessful e-sequences are all caused by only a single unsuccessful trial.

To summarise, several best evolved teams look quite effective in repeatedly solving individual temporal sections of the task (i.e., single trials), even if none of the best teams proved to be 100% successful. One evolutionary run (n. 3) managed to generate several successful and robust teams, with the best of them capable of executing about 80% of successful e-sequences at the post-evaluation test. The performances of best evolved teams from other evolutionary runs are significantly reduced if evaluated with respect to the percentage of successful e-sequences. This data indicates that, for most of the teams, the evolutionary scores have been an overestimation of the effective behavioural capabilities of the teams. We will further discuss the reasons of the post-evaluation performance drop of some of the best evolved teams in the next section.

### Analysis of the best evolved team

In this Section, we show data collected during the post-evaluation test aimed to illustrate some of the features of the best evolved team (i.e., team n. 3). In particular, we show some qualitative data referring to role-switching behaviour. Recall that, an e-sequence is made of 4 trials, during which the robots controller is not reset. In an e-sequence, a team experiences both types of environment in the following order: trial 1 *role F*; trial 2 *role P*; trial 3 *role F*; trial 4 *role P*. Role-switching is a robot's behaviour that happens between two consecutive trials. Within a single trial, a robot can only play either *role P* or *role F*. A role-switching event refers to any change of role from *role P* to *role F* or vice-versa.

How many robots of team n. 3 switch role during an e-sequence? Due to the nature of the task, a team can employ different strategies with respect to role-switching. For example, a strategy with relatively small amount of role-switching behaviour is one in which one robot systemati-

Table 1: Table showing, for each team, num. of unsuccessful e-sequence with failure in: 1 trial (col. 2), 2 trials (col. 3), 3 trials (col. 4), 4 trials (col. 5). Best team in grey.

Team n.	Num. trial failed			
	1	2	3	4
1	14	6	0	0
2	23	11	1	0
3	13	0	0	0
4	8	16	29	9
5	7	20	25	8
6	23	3	2	0
7	22	19	12	2
8	15	29	15	2

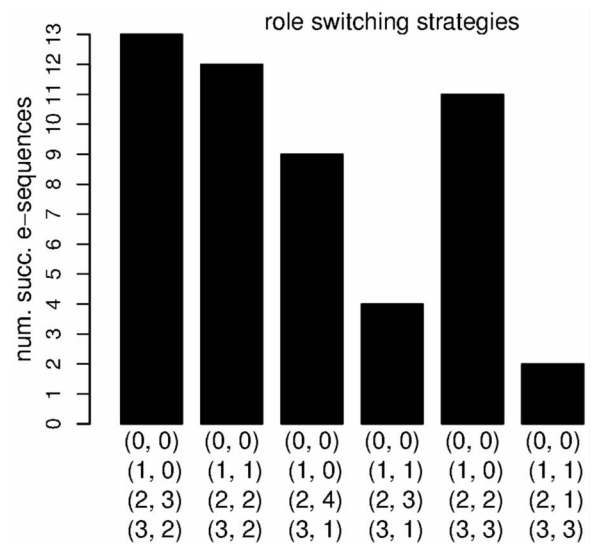


Figure 4: Graphs showing, for team n. 3, the different behavioural strategies (indicated by the four 2-tuples on the x-axis) and their frequencies recorded in the 51 successful e-sequences of the re-evaluation test. On the x-axis, the first element of each 2-tuple refers to the number of role-switching event and the second element refers to the number of robots.

cally switches role at the beginning of a trial (for a total of 3 switches), two robots systematically play *role F*, and two robots systematically play *role P*. A strategy with a significant amount of role-switching behaviour is one in which all the robots change roles in response to any type of environmental change. Fig. 4 shows all the behavioural strategies of team n. 3 observed among all the 51 successful e-sequences recorded at the post-evaluation test. A behavioural strategy is described by 4 2-tuples, in which the first element of each 2-tuple refers to the number of role-switching event and the second element refers to the number of robots. For example, the first bar in Fig. 4 refers to the strategy in which during an e-sequence, 3 robots switch role two times, and 2 robots switch role three times. This is the strategy employed more frequently by team n. 3 during successful post-evaluation e-sequences. However, we can see that this team employs up to 6 different strategies, all of them characterised by the fact that all the robots switch role at least once during a successful e-sequence. The team relies on a variety of strategies which are highly dynamics with respect to role-switching behaviour. Thus, the evolutionary conditions characterised by our experimental scenario are sufficient to generate controllers that are plastic enough to avoid behavioural specialisation (i.e., robots that play only a single role throughout an entire e-sequence).

How much is the role that a robot play determined by the characteristics of the environment? The graph in Fig. 5 shows, for each robot of the best evolved team (n. 3), the

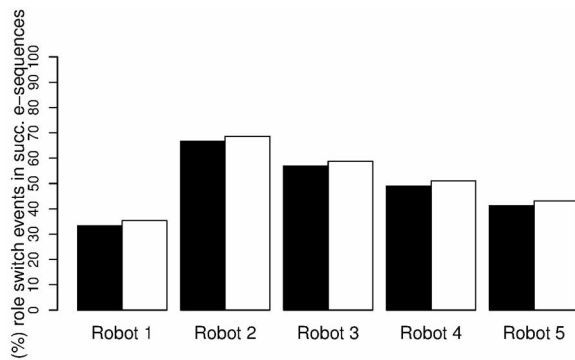


Figure 5: Graph showing, for each robot of team n. 3, the percentage of successful post-evaluation e-sequences, in which the robot plays a different role in different trials presenting the same type of environment. Black bars refers to *Env. A*, white bars to *Env. B*.

percentage of successful e-sequences, at the post-evaluation test, in which the robot plays a different role in trials associated to the same type of environment. This graphs indicates that the type of environment experienced by a robot is not the only cue that determines the role a robot plays within a trial. This is because, for both *Env. A* and *Env. B*, the robots do not necessarily play the same role in different trials associated to the same type of environment. The simple rule “one role for each type of environment” does not systematically apply to any robot of the team n. 3. There is a certain amount of variability among the robots, with robot 1 (i.e., the robot initialised in sector 1, see Fig. 1c) being the most reluctant to play different roles in different trials of an e-sequence presenting the same type of environment (see Fig. 5). Nevertheless, this evidence suggests that the roles are genuinely determined by a combination of factors. In particular, the physical interactions among the robots bring forth contingent phenomena that, biased by the current environmental condition, guide the role-allocation process.

Without further evidence, it is premature to speculate on the operational mechanisms that guides the decision process in this scenario. In similar studies, in which homogeneous teams have been evolved to solve dynamic role-allocation tasks using infra-read sensors as means for communication, the decision making process turned out to heavily rely on the noise injected into the simulation (e.g., Ampatzis et al., 2009). It is likely that, even in our scenario, the robots exploit the noise to break the symmetries due to the homogeneity of the system, and to “diverge” on different roles. However, this scenario differs from those described in the studies above mentioned for two main features: first, the robots are required to repeatedly go through the decision making process, without the robot controllers being reset. Thus, from a functional point of view, the symmetry condition of the robots controller applies only to trial 1. Second,

variable environmental structures and the robots interactions have to equally contribute to the dynamic allocation of roles to robots. This means that, the noise may not be so important in guiding the decision making as it turned out to be in the above mentioned studies.

The relative positions of the robots at the beginning of each trial can potentially be another element that influences the role-allocation process. By visually inspecting the team behaviour, we noticed that most of the best evolved strategies are particularly sensitive to the variability in the initial relative positions of the robots. Too much variability is highly disruptive. Another evidence in favour of the significance of the “initial relative positions” hypothesis for the role allocation process can be found in Fig. 3. The graphs shows that the best three teams (n. 1, 3, 6) did worst in the first temporal section (trial) than in the following three sections at the post-evaluation test (compare black with all the other bars in Fig. 3, for team n. 1, 3, 6). Recall that, in trials following the first one, the robots can position themselves, through their movement, in a way to facilitate the execution of the task. In the first trial of each e-sequence the robots are pseudo-randomly positioned. The robots’ positioning algorithm has been intentionally designed to introduce some variability into the system. However, it seems that, even the best evolved strategy can only cope with a limited portion of this variability. Future work and analysis on the operational mechanisms used by the robots to allocate role may shed light on whether and the extent to which the robots relative position at the beginning of the trial has an impact on the performance of the team, and on the role that each robot play within the team. However, it seems plausible to think that the sensitivity of the best evolved teams to the variability in the robots’ initial relative position can account for the fitness drop observed between evolution and post-evaluation.

## Conclusion

In this study, we have investigated the evolutionary conditions that facilitate the emergence of role-allocation and role-switching behaviour in teams of homogeneous robots. To the best of our knowledge, this is the first simulated study in which: i) role-allocation and role-switching behaviour are both required for the benefit of the team. ii) role-allocation and role-switching behaviour are evolved in team of more than 3 agents; iii) a relatively complex team’s behaviour, based on a different distribution of robots-to-roles, is obtained using only infra-red sensors as means for interaction.

We consider this study the first step towards the development of swarm robotics models that could shed light on the evolution of self-organised role-allocation and role-switching behaviour. Similarly to other swarm robotics studies, we are motivated by engineering and biological objectives. From an engineering perspective, our objective is to look at task/role-allocation and task/role-switching behaviour in order to generate design principles that preserve

the adaptability and flexibility of both the system components and of the resulting processes. With respect to this, the results of this study demonstrate that a task requiring complex group response in simulated agents with limited sensory and communication capabilities can be solved by teams of homogeneous robots controlled by dynamic neural network synthesised by artificial evolution.

From a biological perspective, we aim at generating new insights into the general properties of large scale distributed natural systems. In particular, this project aims at providing a principled understanding of the temporal development of task/role-allocation and task/role-switching behaviour by designing models that look at the effect of evolution on operating and design principles. Further work and analysis of the working principles of the evolved solutions is certainly required to be able to understand how our simulated teams solve the task, and how this can help us to understand natural swarms. However, at first glance, we can already mention a couple of phenomena that seem to play a significant role for the evolution and development of self-organised role-allocation and role-switching behaviour. The first element is the way in which the robots access and leave the nest. In our scenario, the robots have been left free to move in and out the nest from any directions. However, the best evolved strategies are based on movements which induce all the foragers to exit and re-enter the nest from a specific limited area. This suggests that structural properties of the nest may interfere and maybe facilitate the adaptive re-distribution of agents-to-roles. Another element is the amount of environmental variability the system can cope with. Our robots, which communicated only through infra-red sensors, bring forth strategies that seem quite fragile with respect to various sources of variability, such as the cardinality of the team, the distance between nest and foraging site, the length of a trial, the order in which the different environmental conditions are experienced, etc. Future work will concentrate on the investigation of alternative means of communication that could strengthen the effectiveness of the self-organised re-distribution of agents-to-roles.

## References

- Ampatzis, C., Tuci, E., Trianni, V., Christensen, A., and Dorigo, M. (2009). Evolving self-assembly in autonomous homogeneous robots: Experiments with two physical robots. *Artificial Life*, 15(4):465–484.
- Anderson, C., Franks, N., and McShea, D. (2001). The complexity and hierarchical structure of tasks in insect societies. *Animal Behaviour*, 62:643–651.
- Beer, R. D. and Gallagher, J. C. (1992). Evolving dynamic neural networks for adaptive behavior. *Adaptive Behavior*, 1(1):91–122.
- Bongard, J. (2000). The legion system: A novel approach to evolving heterogeneity for collective problem solving. In Poli, R., editor, *Genetic Programming: Proc. of EuroGP-2000*, LNCS, pages 16–28. Springer-Verlag.
- Camazine, S., Deneubourg, J.-L., Franks, N., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2001). *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ.
- Davidson, D. (1977). Foraging ecology and community organization in desert seed-eating ants. *Ecology*, 58(4):725–737.
- Dorigo, M. and Şahin, E. (2004). Guest editorial. Special issue: Swarm robotics. *Aut. Rob.*, 17(2–3):111–113.
- Duarte, A., Weissing, F., Penn, I., and Keller, L. (2011). An evolutionary perspective on self-organised division of labour in social insects. *Annual Review of Ecology, Evolution, and Systematics*, 42:91–110.
- Eiben, A., Nitschke, G., and Schut, M. (2007). Collective specialization for evolutionary design of a multi-robot system. In Sahin, E., Spears, W., and Winfield, A., editors, *Proc. of the 2<sup>nd</sup> Int. Workshop on Swarm Robotics*, volume 4433, pages 189–205. Springer.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.
- Gordon, D. (1996). The organisation of work in social insects. *Nature*, 380:121–124.
- Ijspeert, A., Martinoli, A., Billard, A., and Gambardella, L. (2001). Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Aut. Rob.*, 11(2):149–171.
- Jakobi, N. (1997). Evolutionary robotics and the radical envelope of noise hypothesis. *Adaptive Behavior*, 6:325–368.
- Labella, T., Dorigo, M., and Deneubourg, J.-L. (2006). Division of labour in a group of robots inspired by ants’ foraging behavior. *ACM TAAS*, 1(1):4–25.
- Luke, S. and L.Spector (1996). Evolving teamwork and coordination with genetic programming. In *Genetic Programming 1996. Proc. of the 1<sup>st</sup> Annual Conf.*, pages 150–156. MIT Press, Cambridge, MA.
- Nolfi, S. and Gigliotta, O. (2010). Evorobot\*. In Nolfi, S. and Mirolli, M., editors, *Evolution of Communication and Language in Embodied Agents*, pages 327–332. Berlin-Heidelberg: Springer-Verlag.
- Panait, L. and Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11:387–434.
- Quinn, M. (2001). A comparison of approaches to the evolution of homogeneous multi/robot teams. In *CEC*, volume 1, pages 128–135.
- Quinn, M., Smith, L., Mayley, G., and Husbands, P. (2003). Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors. *Phil. Trans. of the Royal Society of London, Series A*, 361:2321–2344.
- Tuci, E. and Trianni, V. (2012). On the evolution of homogeneous multi-robot teams: clonal versus aclonal approach. In Ziemke, T., Balkenius, C., and Hallam, J., editors, *SAB XII*, volume 7426 of *LNAI*, pages 391–400. Springer.
- Vassie, K. and Morlino, G. (2012). Natural and artificial systems: Compare, model or engineer? In Ziemke, T., Balkenius, C., and Hallam, J., editors, *SAB XII*, volume 7426 of *LNAI*, pages 1–11. Springer.



# Tipping points in Complex Coupled Life-Environment Systems

Iain S. Weaver\* and James G. Dyke

School of Electronics and Computer Science, University of Southampton, UK, SO17 1BJ

\*isw1g10@soton.ac.uk

## Abstract

Simple models of complex phenomena provide powerful insights and suggest low-level mechanistic descriptions. The Earth system arises from the interaction of subsystems with multi-scale temporal and spatial variability; from the microbial to continental scales, operating over the course of days to geological time. System-level homeostasis has been demonstrated in a number of conceptual, artificial life, models which share the advantage of a thorough and transparent analysis. We reintroduce a general model for a coupled life-environment model, concentrating on a minimal set of assumptions, and explore the consequences of interaction between simple life elements and their shared, multidimensional environment. In particular stability, criticality and transitions are of great relevance to understanding the history, and future of the Earth system. The model is shown to share salient features with other abstract systems such as Ashby's Homeostat and Watson and Lovelock's Daisyworld. Our generic description is free to explore high-dimensional, complex environments, and in doing so we show that even a small increase in the environmental complexity gives rise to very complex attractor landscapes which require a much richer conception of critical transitions and hysteresis.

## Introduction

The principle that environmental factors affect life is evident throughout the biosphere both regionally and globally, and throughout Earth's history (Gaston 2000). Variables such as temperature and soil or atmospheric composition determine whether an organism can proliferate, and populations will develop to some resource limited level. Climatic shifts between, for example, greenhouse and icehouse states are accompanied by mass extinction (Haywood 2004). Indeed this principle is at the core of environmental niche modelling, where the distribution of species in the space of significant environmental factors, their *realized-niche*, is used to predict the spatial species distribution (Thomas et al. 2004). The influence of life on its environment however, and whether it is expected to have a stabilising, or destabilising effect is less clear. Lovelock and Margulis's (1974) original "Gaia Hypothesis" focused on the extent to which the emergence of life has promoted a self-regulating, or homeostatic system. In this work we consider the ability of a coupled life-

environment system to generate stabilising feedback loops in increasingly complex environments under a minimal set of assumptions.

The Gaia-hypothesis proposes that life may self-organise into complex, self-regulating systems with maintain the habitability of their environment and at the higher level contribute to regulation of variables globally conducive for life to flourish. Controversy followed as it is unclear how regulatory mechanisms might emerge without the need for system-wide cooperation which would contradict the principle of natural selection at the species level (Doolittle 1981). Watson and Lovelock's (1983) "Daisyworld" took the first step towards addressing this with an abstract coupled life-environment model whose biota consists of two species of daisy which exert a unidirectional force on their shared environment, reduced to a single temperate variable. Black daisies absorb a large amount of energy, and have a warming effect compared to white daisies which have a high albedo and an overall cooling effect. If the species are organised such that black daisies out-compete white daisies at low temperatures, the model shows that infinitesimally differentiated species can establish *reign-control* over their environment (Harvey 2004); a small reduction in global temperature allows the black absorptive daisies to proliferate and visa-versa for increases in temperature. In this way, each species holds a reign, opposing changes in one direction such that the planetary temperature is not just stable, but robust to a range of external perturbations. There have been a number of extensions and developments of the original Daisyworld model (see Wood et al. (2008) for a review), some of which has been undertaken within the field of artificial life. Dyke et al. (2007) for example allow external perturbation to vary on timescales comparable to changes in the biota, while Williams and Noble (2005) extend the model to enable stochastic evolution of daisy species.

We aim to address two short-comings of this model in detail. Firstly, while useful in its transparency, the behaviour of such a one-dimensional dynamical system is extremely limited. Iconic phenomena such as hysteresis loops are a common metaphor for transitions in very complex systems,

although do not take into account the additional degrees of freedom of high dimensional systems which may exhibit cyclical, complex or chaotic behaviour, inaccessible to such a simple metaphor. It is therefore unclear the extent to which this picture fits such high-dimensional systems. On the other hand, while more complicated many-body systems benefit from a much richer zoo of emergent behaviours, it typically comes at the expense of transparency, exchanging generality for a more faithful representation of a specific system. Along with increasingly complex environments, the relevance of the reign-control mechanism to very much larger populations of diverse biotic elements is unclear. Daisyworld does not address the mechanism by which the pair of antagonistic species might emerge. While later work makes clear that the mechanism is not unique to a two-species model, a persistent theme of such models is that they are in some sense *designed* (Lenton 1998).

Along with these key points, we explore the nature of critical transitions or “regime shifts” (Williams and Lenton 2010). We utilize a Daisyworld-type model first introduced in Dyke (2010) then extended in Dyke and Weaver (2013) in order to assess to what extent iconic phenomena such as hysteresis loops, which are a common metaphor for transitions in very complex systems, are appropriate when considering higher dimensional systems. We will show that even a small increase in the environmental complexity generates surprisingly detailed, and complex attractor landscapes. Understanding the way in which we traverse the space of environmental variables calls for a richer conception of critical transitions and hysteresis. In short we find that some critical transitions have little impact on system behaviour while others can produce large changes from states which may not be recoverable, resulting in asymmetrical transitions. Importantly, there appears no immediate way in which to differentiate between them and thus our results may be interpreted as urging caution for the use of early warning signals for complex systems.

### Model formulation

Dyke’s (2010) “Daisystat” model in single environmental variable mode can be understood as a simplified Daisyworld model, very similar to the Artificial Life model of McDonald-Gibson et al. (2008). A population of simple organisms is affected by their environment such that they thrive in the vicinity of their ecological niche, and in turn influence the environment through simple, linear feedbacks. The key differences between the Daisystat model, and Daisyworld are that Dyke’s (2010) enables the description of higher dimensional environments and rather than prescriptive feedbacks, the model employs random life-environment interactions.

Throughout this work, we describe the distribution of biota through  $K$  variables,  $\alpha$ , denoting the overall activity of individual elements, or populations, influenced by the state

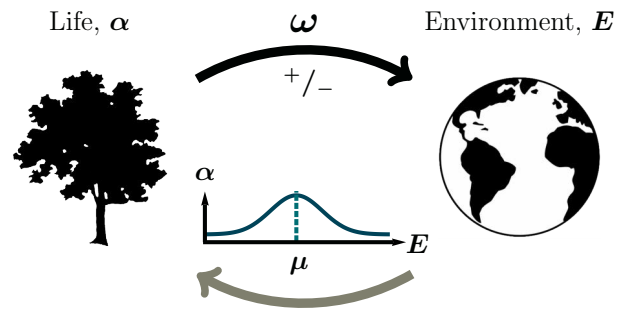


Figure 1: In the simplest case, biotic elements,  $\alpha$ , have an increasing or decreasing,  $\omega$ , effect on their environment  $E$ . In turn, they are only abundant over a finite range of the environmental variables, centred on their fundamental niche,  $\mu$ .

of their shared environment, represented the  $N$  variables in the vector  $E$ , where we have used boldface notation to denote vectors, and will reference individual elements with subscripted indices.

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}$$

The first principle assumption is that components of the biota are only significantly abundant, or active, in the vicinity of their ideal environmental conditions, or *niche*. The second assumption is that the environment is itself influenced by the biota; environmental variables may be decreased or increased by the individual biotic elements, through consumption, excretion or some other process with no bias towards positive or negative feedback. In essence, the model consists of these two principle assumptions, illustrated by Fig. 1. This section elaborates on these assumptions, along with their consequential behaviour.

**i) Environment affects life** Each element of the biota only has a significant presence in a narrow range of environmental conditions wherein it may proliferate, respire, or is otherwise active. As environmental conditions depart this niche, the activity is reduced until the population recedes or becomes dormant such that its presence is negligible. While many species may occupy a wide range of niches, we may add an additional constraint that there is a limited range of environmental conditions which are conducive to life at all, such as the conditions to maintain liquid water. This confines the niche for any population to some volume of the space of environmental variables within which life may exist, known as the *essential range* (Ashby 1952), given by the range  $[0 : R]$  for simplicity.

A simple choice of function to describe the changes in the

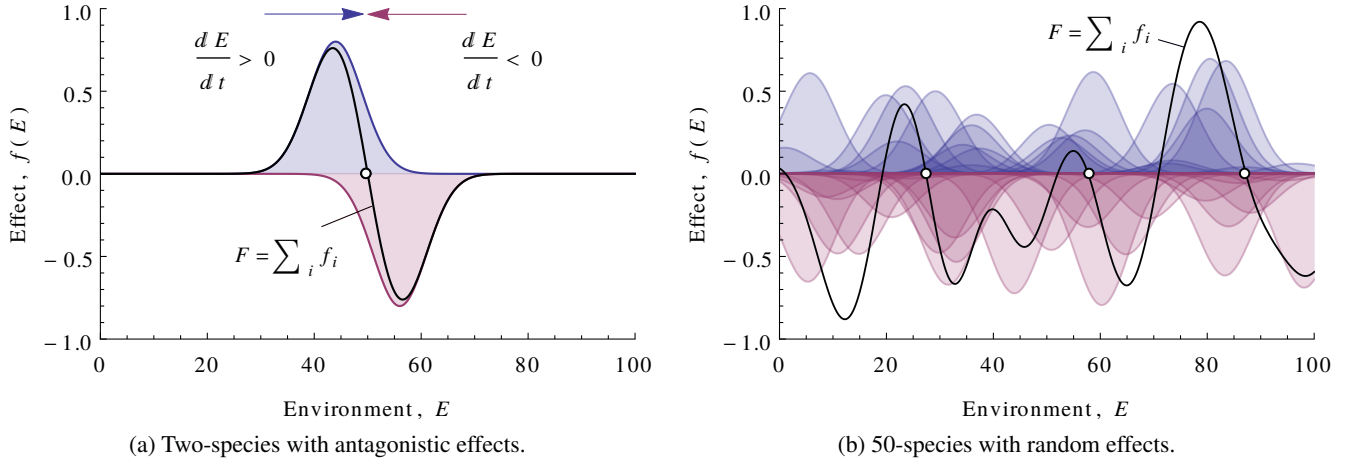


Figure 2: Activity of individuals with increasing (blue) and decreasing (red) effects on their environment, along with the net effect  $F(E) = \sum_i f_i(E)$ . Fixed points, shown by open circles, occur where the sum of individual effects is exactly zero. Fig. 2a is analogous to Daisyworld and illustrates the reign-control mechanism, along with a single fixed point. Fig. 2b shows large numbers of species with random effects (with no preference for negative, or positive feedback), demonstrating the emergence of such points *by chance*.

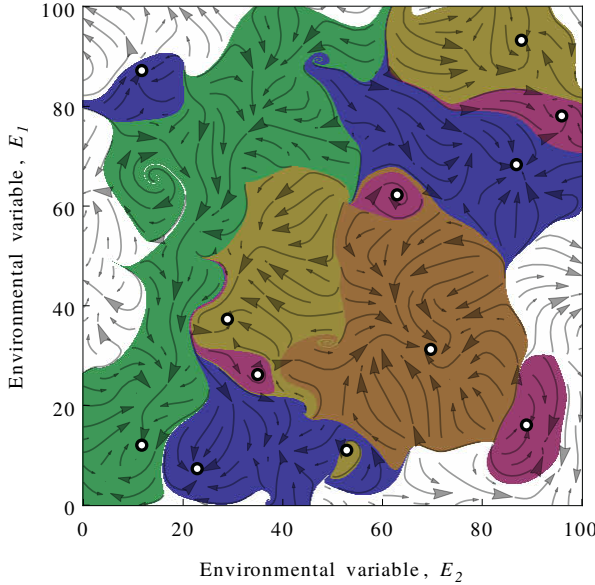


Figure 3: The two environmental variable model with no external perturbation,  $\mathbf{P} = \mathbf{0}$ . Points indicate the position of stable fixed points, where  $\mathbf{F}(\mathbf{E}) = \mathbf{0}$  and  $\nabla \mathbf{F} < \mathbf{0}$ . Shaded regions show the basin of attraction for each point indicating that a model initiated within the region will arrive at the fixed point. Increasingly positive or negative perturbations will influence the shape of the attractor space such that some attractors will disappear, and new attractors may even emerge, resulting in rapid transitions.

biota would be a linear relaxation towards some steady-state;

$$\tau_\alpha \frac{d\alpha_i(t)}{dt} = \alpha^*(\mathbf{E}, \boldsymbol{\mu}_i) - \alpha_i(t) \quad (1)$$

where  $\tau_\alpha$  defines the timescale of changes in the population  $\alpha_i(t)$  towards its steady-state value,  $\alpha^*(\mathbf{E}, \boldsymbol{\mu})$  is the steady-state distribution about the individual niche  $\boldsymbol{\mu}_i$ . It has been shown that the specific choice of this distribution is unimportant providing it has a well defined variance (Dyke and Weaver 2013). In this instance we choose a Gaussian distribution with characteristic width  $\sigma_E$ , centred on  $\boldsymbol{\mu}$  chosen randomly in the interval  $[0 : R]$ , the essential range.

$$\alpha^*(\mathbf{E}, \boldsymbol{\mu}) = \exp\left(-\frac{|\mathbf{E} - \boldsymbol{\mu}|^2}{2\sigma_E^2}\right) \quad (2)$$

**ii) Life affects environment** On the other hand, the biota has an effect on their shared environment. Individual populations may modify environmental variables independently, having either an increasing or decreasing effect. Black daisies in Watson and Lovelock's (1983) Daisyworld for example absorb a large amount of radiation compared to white daisies, having an increasing effect on temperature. In the simplest case, the biota has an effect,  $\mathbf{f}$ , in proportion to their activity

$$\mathbf{f}_i(t) = \omega_i \alpha_i(t) \quad (3)$$

where  $\omega_i$  is the effect of an individual on the environment  $\mathbf{E}$ , chosen randomly from the interval  $\pm 1$ . The net effect of the biota on the environment,  $\mathbf{F}$ , is simply found by summing individual contributions

$$\mathbf{F}(t) = \sum_{i=1}^K \mathbf{f}_i(t). \quad (4)$$

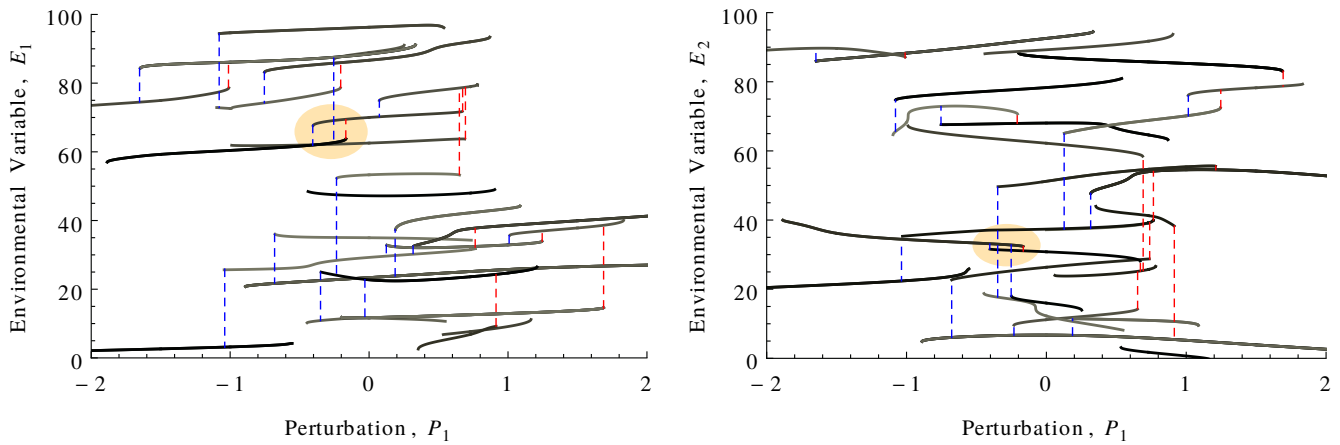


Figure 4: The states and transitions which exist in a two-dimensional environment model for a range of  $P_1$ , where  $P_2 = 0$ . States are shaded randomly to help differentiate them, and end abruptly where states become unstable and transition outside the essential range, as opposed to towards another stable state. Transitions to other states are colored red and blue where the transition is encountered by increasing and decreasing perturbation respectively. Importantly, not all transitions are symmetrical. Several transitions occur from states which may not be recovered by trivially applying an opposing perturbation as would be expected in the case of a one-dimensional system. Indeed some states undergo no transitions at all, besides those which would drive the system outside the essential range. The basins of attraction which lead to the hysteresis loop marked are represented by Fig. 5.

Additionally, each of the environmental variables may be affected by some external abiotic perturbing force,  $\mathbf{P}$ . In the original Daisyworld model, the single environmental variable was temperature, and the perturbing force was insolation, the influence of Daisyworld's star. Other perturbing effects may include flux of chemicals or gasses, such as by volcanic or anthropogenic emission. The net change in the environment is the sum of these features

$$\tau_E \frac{d\mathbf{E}(t)}{dt} = \mathbf{P} + \mathbf{F}(t) \quad (5)$$

where  $\tau_E$  is the characteristic timescale for changes in environmental variable  $i$ , chosen to be equal between variables for convenience, and  $F_i$  is the sum of effects from the biota.

It is important to establish the relationship between the times scales of changes in the biota,  $\tau_\alpha$ , and their shared environment,  $\tau_E$ . Progress towards establishing the behaviour of such models has been demonstrably simpler by assuming a separation of timescales  $\tau_E \gg \tau_\alpha$ . In the Daisyworld model for example, this leads directly to an analytical solution (Weaver and Dyke 2012). In this limit, the populations  $\alpha_i(t)$  quickly adjust to their steady-state values  $\alpha_i^*(\mathbf{E}, \boldsymbol{\mu}_i)$ , removing the time dependence in  $\mathbf{F}(t) \rightarrow \mathbf{F}(\mathbf{E})$ . Furthermore, the summed effect  $\mathbf{F}$  is normalised to have a variance  $\sigma_F^2 = 1$  for convenience.

Similarities with Watson and Lovelock's (1983) Daisyworld are clear for simple, one-dimensional environments. Fig. 2 illustrates the individual effects of the individual populations,  $f_i$ , along with the total effect,  $F$ . Fig. 2a shows the Daisyworld control mechanism, so called reign-control.

The two populations exert opposing forces on their environment, where the population with a positive effect is abundant at low ranges, while the negative effect dominates for larger values. This system naturally finds stability at a point where these effects are exactly in balance, somewhere between the niches of the opposing populations. Fig. 2b shows 50 *random* populations (random in both their niche, and influence of the environment). Importantly, fixed points occur where the net effect,  $F$ , crosses zero, and are furthermore stable only when the gradient is negative. In the case of Daisyworld, it may be remarked that the single fixed point is prescribed in the model formulation. However, previous work has concentrated on the emergence of these points from interactions between many populations and their shared environment *by chance*, finding them to be a generic property of such models, largely independent of the dimensionality of  $\mathbf{E}$  (Dyke and Weaver 2013).

## Higher Dimensions

Typically, illustrations of hysteresis involve a single variable. Such examples however bely the much more colourful behaviour which emerges in higher dimensional systems. Firstly, it has been shown that the expected number of fixed points increases exponentially with the dimensionality of the environment (Dyke and Weaver 2013), resulting in very many more stable states. This may be contrary to intuition as a fixed point must be stationary in all dimensions simultaneously, and appears exponentially unlikely with more complex environments. However, this is opposed by the increase



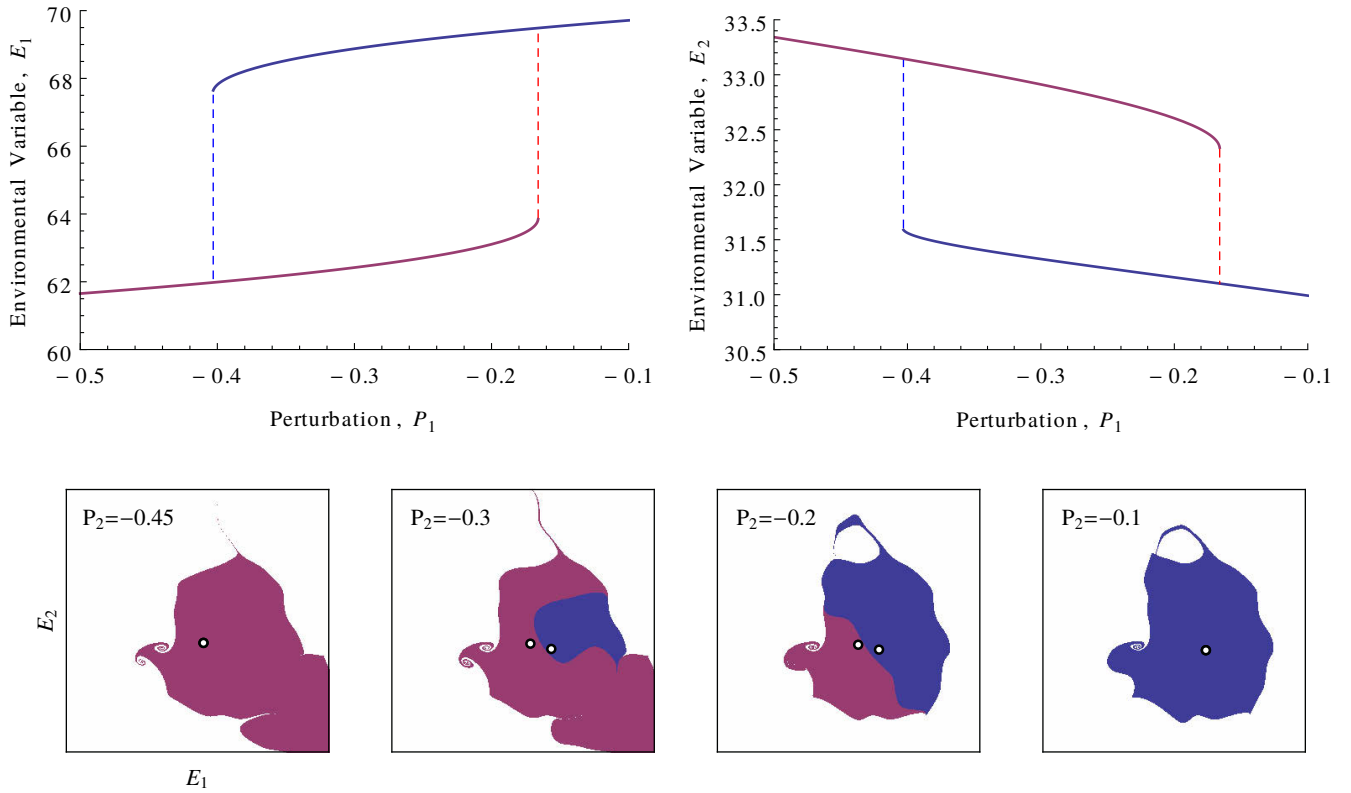


Figure 5: Colored regions correspond to the basins of attraction of the two fixed points (shown by open circles) highlighted in Fig. 4. As the perturbation is varied, one fixed point vanishes, and its basin of attraction is encompassed by that of the other fixed point, resulting in a transition. In this instance, the transition is roughly symmetrical and the previous state may be recovered by reversing the perturbation.

in volume of the environment  $E$ , which enables many more configurations of the biota. Here, we demonstrate the complicated network of transitions between these states with increasing or decreasing perturbations. Fig. 3 shows a randomly initialised two-dimensional system of fixed points for the model subjected to no external perturbation. The response of these fixed points to perturbation is demonstrated by Fig. 4, showing the same two-dimensional system for increasing and decreasing perturbations in one direction. The complicated behaviour is perhaps best interpreted in terms of attractors, where transitions occur when a basin of attraction is fully succeeded by another, illustrated by Fig. 5 for a roughly symmetrical transition between two states. Some notable features of Fig. 4 include

**Density of states** The stable states consistent with a particular driving force exhibits great diversity. Rather than bistability, Fig. 4 shows over a dozen stable states at a given perturbation. Furthermore Dyke and Weaver (2013) finds the density of stable states increases exponentially with environmental complexity. It is not unreasonable to expect higher dimensional environments to house hundreds of vi-

able stable states. Another interesting observation of Fig. 4 is that the density of stable states diminishes (exponentially) with increasing perturbation strength. In a one dimensional system for example, the mean number of stable states in a unit interval is given by

$$n = \frac{1}{2\sqrt{2\pi}\sigma} \exp\left(-\frac{P^2}{2}\right) \quad (6)$$

while is illustrated in Fig. 6. It is intuitive that the summed contributions of random, uncorrelated effects is Gaussian distributed, and therefore the density of points with a strong enough effect to oppose increasing perturbations decreases.

**Step size** The geological record is punctuated by abrupt and in some sense catastrophic transitions (Alley et al. 2003). Such events correspond to large, quantitative changes although in a high dimensional space it is important to note that such steps may not necessarily be large in all dimensions. Fig. 4 illustrates that many transitions in which one or the other environmental variable undergoes a very small change. This observation highlights the importance in identifying the significant axes of change in understanding high

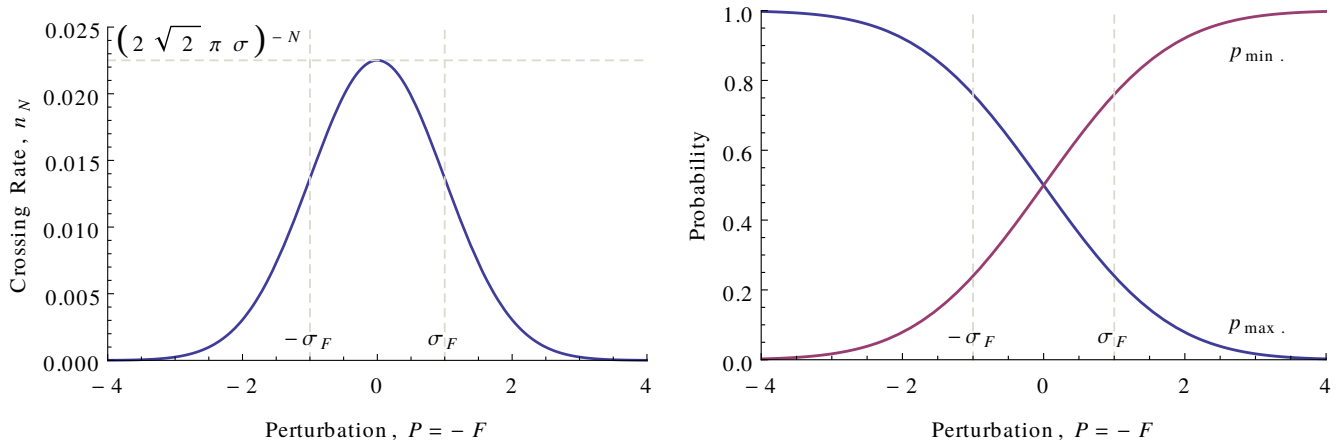


Figure 6: Eq. (6) (left) shows the density of fixed points over a range of perturbations to be Gaussian distributed, such that exponentially fewer stable states are available in the face of large perturbations. Additionally we can derive the probability of a transition occurring in the increasing or decreasing  $P$ -direction as a function of  $P$  (right). While extrema are equally likely at any value of  $P$ , they are more likely to be maxima where  $P$  is negative, and minima where it is positive. Transitions encountered by increasing  $P$  are therefore more likely when  $P$  is positive and visa-versa. The width of the functions is related to the variance of  $F$ , which is normalised to  $\sigma_F^2 = 1$ .

dimensional transitions as opposed to examining a lower dimensional projection. It is also interesting to examine the probability that a transition occurs in the direction of increasing perturbation (local minima of  $F(E)$ ) or in the decreasing direction (local maxima). To achieve this, we examine the correlations between  $F$  and it's second derivative,  $F''$  by formulating the covariance matrix, the result of which is shown in Fig. 6. This illustrates that positive forcing is more likely to encounter a local minimum, and therefore a transition caused by increasing  $P$  while the reverse is true for negative forcing.

**Reversibility** The one-dimensional hysteresis metaphor describes a symmetrical loop, emphasising only that systems may be bistable over a range of external perturbation, and that sufficiently strong driving can move the system between states. However, while Fig. 5 shows such transitions exist, the two dimensional picture significantly obfuscates this as along with simple hysteresis loops, we see large numbers irreversible regime shifts. This occurs where driving in a particular environmental variable will not revert the system to its original state. This result would have particularly important consequences to real-world transitions. Transitions in lakes to eutrophic states for example reduce biodiversity and cause difficulty in water treatment (Wang et al. 2012). Reversing these transitions is a key concern, and this model emphasises the importance of a thorough understanding of the system in question. A system collapsed by strong driving in one dimension may not be recoverable simply by reversing the effect.

## Discussion

We have shown that self-regulation is a mechanism which may arise from a large population of random life elements, and explored this mechanism in a multi-dimensional environment. In particular, we note three important differences between the one-dimensional picture of hysteresis between bistable states; the magnitude of a regime-shift following a transition is highly variable, returning to a previous state after a transition may be impossible, and the density of viable states for a given perturbation is exponentially large with the number of environmental variables. These points prompt a number of questions when considering transitions in climatic or ecosystems; which aspects of the environment if any are expected to undergo catastrophic changes? Can the previous state be recovered, and if so, which are the important dimensions of control? How many alternative states are consistent with external forces, and therefore stable?

Two main questions are left open in our implementation. While it has been shown that the choice of function for the steady state populations about their niche is unimportant, a factor which would convolute this such as interspecies competition are not resolved. In this sense, we do not distinguish between the fundamental niche, the environmental conditions in which the species can survive and proliferate, and realized niche which is influenced by external factors such as interspecies competition and predation. While we mediate all biotic interactions through the environment, previous studies have concentrated on the individual species interactions, such as McDonald-Gibson et al. (2008) and Dyke et al. (2007) where increased interspecies competition appears to accentuate the homeostatic properties of a reign-control sys-

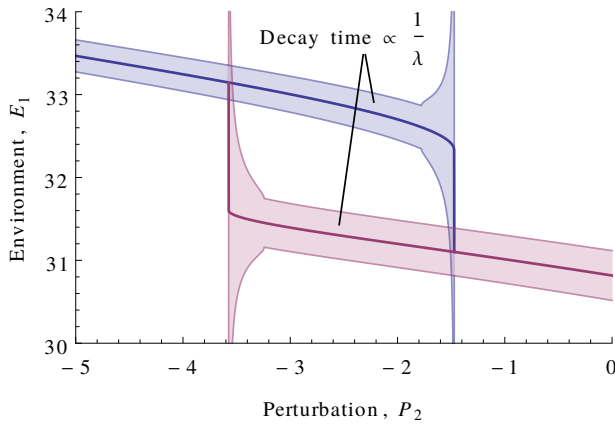


Figure 7: Shaded regions indicate the time taken for the system to recover from small perturbations from the fixed points of Fig. 5. The decay time is inversely proportional to the real part of the largest eigenvalue of the Jacobian evaluated at the fixed point. The discontinuity is caused by the transition between an over-damped system, where eigenvalues are real and perturbations decay exponentially, and an under-damped system where complex eigenvalues result in damped oscillations.

tem, despite other approaches which argue that increasingly connected systems lose stability (May 1972).

Also, the extent to which the separation of time scales between changes in the environment and biota are important to the model dynamics is unclear. While it has no influence on the model fixed points, illustrations of basins of attraction, such as Fig. 5 are expected to change significantly, and there is no clear way to represent them in a  $N + K$  dimensional system. However, some progress may be possible by assuming the simplest case of linear relaxation of populations towards some steady state value. Starting from Eq. (1), it can be shown that such a case removes the need to resolve the individual populations  $\alpha_i$ , reducing to a  $2N$  dimensional system. Multiplying by  $\omega_i$  and summing over the population gives

$$\tau_\alpha \sum_i \omega_i \frac{d\alpha_i(t)}{dt} = \sum_i \omega_i \alpha_i^*(\mathbf{E}, \boldsymbol{\mu}_i) - \sum_i \omega_i \alpha_i(t)$$

$$\tau_\alpha \frac{d\mathbf{F}(t)}{dt} = \mathbf{F}^*(\mathbf{E}) - \mathbf{F}(t) \quad (7)$$

by the definition for  $\mathbf{F}$  of Eq. (4). Previous work has shown that the relationship between the time scales of changes in the biota and changes in external perturbation set limits on the ability of a system to self-regulate, and invite a range of new phenomena to emerge (Weaver and Dyke 2012).

Critical slowing-down (CSD) (Lenton 2011) refers to the long relaxation time of near-critical systems, those which are approaching climatic or ecosystem transitions. As natural and anthropogenic pressures stress aspects of the Earth

system, it has been shown that certain time series find an increase in the auto-correlation coefficient precedes such transitions (Dakos et al. (2008) analyze eight ancient abrupt climate shifts). This signal suggests that the relaxation of the system towards its steady state slows as it approaches a regime shift. Fig. 7 illustrates the decay time of the system shown in Fig. 5 and verifies that the type of system described in this work exhibits this signal. The eigenvalues of the Jacobian in the vicinity of a fixed point yield information not only pertaining to the stability of the point, but also provide estimates of the decay time of small fluctuations from the fixed point. A large decay time indicates the system is in some sense *slow*. However it appears to possess no further information pertaining to the direction, magnitude or reversibility of the transition; clearly important and relevant questions when considering transitions in real systems.

## Acknowledgements

This work was supported by an EPSRC Doctoral Training Centre grant (EP/G03690X/1).

## References

- Alley, R. B., Marotzke, J., Nordhaus, W., Overpeck, J., Peteet, D., Pielke, R., Pierrehumbert, R., Rhines, P., Stocker, T., Talley, L., et al. (2003). Abrupt climate change. *science*, 299(5615):2005–2010.
- Ashby, W. (1952). *Design for a brain*. Wiley.
- Dakos, V., Scheffer, M., van Nes, E. H., Brovkin, V., Petoukhov, V., and Held, H. (2008). Slowing down as an early warning signal for abrupt climate change. *Proceedings of the National Academy of Sciences*, 105(38):14308–14312.
- Doolittle, W. F. (1981). Is nature really motherly. *CoEvolution Quarterly*, 29:58–63.
- Dyke, J., McDonald-Gibson, J., Di Paolo, E., and Harvey, I. (2007). Increasing complexity can increase stability in a self-regulating ecosystem. In *Advances in Artificial Life*, pages 133–142. Springer.
- Dyke, J. G. (2010). The daisystat: A model to explore multidimensional homeostasis. In *Artificial Life XI, Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 349–359. MIT Press, Cambridge MA.
- Dyke, J. G. and Weaver, I. S. (2013). The emergence of environmental homeostasis in complex ecosystems. *PLoS computational biology*, 9(5):e1003050.
- Gaston, K. J. (2000). Global patterns in biodiversity. *Nature*, 405(6783):220–227.
- Harvey, I. (2004). Homeostasis and rein control: From daisyworld to active perception. In *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems, ALIFE*, volume 9, pages 309–314.
- Haywood, A. M. (2004). From greenhouse to icehouse: The marine eocene-oligocene transition. *Antarctic Science*, 16(4):585–586.

- Lenton, T. M. (1998). Gaia and natural selection. *Nature*, 394(6692):439–447.
- Lenton, T. M. (2011). Early warning of climate tipping points. *Nature Climate Change*, 1(4):201–209.
- Lovelock, J. E. and Margulis, L. (1974). Atmospheric homeostasis by and for the biosphere. *Tellus Series B-Chemical and Physical Meteorology*, 26(4):299–327.
- May, R. M. (1972). Will a large complex system be stable? *Nature*, 238:413–414.
- McDonald-Gibson, J., Dyke, J., Di Paolo, E., and Harvey, I. (2008). Environmental regulation can arise under minimal assumptions. *Journal of theoretical biology*, 251(4):653–666.
- Thomas, C. D., Cameron, A., Green, R. E., Bakkenes, M., Beaumont, L. J., Collingham, Y. C., Erasmus, B. F., De Siqueira, M. F., Grainger, A., Hannah, L., et al. (2004). Extinction risk from climate change. *Nature*, 427(6970):145–148.
- Wang, R., Dearing, J. A., Langdon, P. G., Zhang, E., Yang, X., Dakos, V., and Scheffer, M. (2012). Flickering gives early warning signals of a critical transition to a eutrophic lake state. *Nature*, 492(7429):419–422.
- Watson, A. and Lovelock, J. (1983). Biological homeostasis of the global environment: the parable of daisyworld. *Tellus B*, 35(4):284–289.
- Weaver, I. S. and Dyke, J. G. (2012). The importance of timescales for the emergence of environmental self-regulation. *Journal of Theoretical Biology*, 313(0):172 – 180.
- Williams, H. and Noble, J. (2005). Evolution and the regulation of environmental variables. In *Advances in Artificial Life*, volume 3630, pages 332–341. Springer Berlin Heidelberg.
- Williams, H. T. and Lenton, T. M. (2010). Evolutionary regime shifts in simulated ecosystems. *Oikos*, 119(12):1887–1899.
- Wood, A. J., Ackland, G. J., Dyke, J. G., Williams, H. T. P., and Lenton, T. M. (2008). Daisyworld: a review. *Reviews of Geophysics*, 46:RG1001.



# Quantifying Political Self-Organization in Social Media. Fractal patterns in the Spanish 15M movement on Twitter

Miguel Aguilera<sup>1,3\*</sup>, Ignacio Morer<sup>1,3</sup>, Xabier E. Barandiaran<sup>2,3</sup> and Manuel G. Bedia<sup>1</sup>

<sup>1</sup>ISAAC, Dept. of Informatics, Universidad de Zaragoza, Spain.

<sup>2</sup>IAS-Research Centre for Life, Mind, and Society & Dept. of Philosophy & University School of Social Work, UPV/EHU University of the Basque Country, Spain.

<sup>3</sup>DatAnalysis15M Research Network

\* miguel.academic@maguilera.net

## Abstract

The objective of this work is to better analyse and understand social self-organization in the context of social media and political activism. More specifically, we centre our analysis in the presence of fractal scaling in the form of  $1/f$  noise in different Twitter communication networks related to the Spanish 15M movement. We show how quantitative indexes of brown, white and pink noise correlate with qualitatively different forms of social coordination of protests: rigidly organized protests (brown noise), reactive-spontaneous protests (white noise) and complex genuinely self-organized protests (pink noise). In addition, pink noise processes present correlations that reach much further in time, maintaining a dynamical coherence that last several days, and also show a balance between mean distance and clustering coefficient within the interaction network.

## Introduction

Artificial Life models have helped identify processes of self-organization in the living domain, opening up our scientific imagination and improving our theoretical insight into complex phenomena, ranging from chemical autocatalysis (Kauffman, 1993) to emergent collective intelligence (Bonabeau et al., 1999; Holland and Melhuish, 1999). Human social *life* has also been approached through Artificial Life techniques and theoretical lenses (Hemelrijk and Kunz, 2003) but it is now, with the rise of social media and digital data-mining, that a new door is opened for a genuine analysis and synthesis of human social life (Lazer et al., 2009).

One of the central topics of Artificial Life modelling has been the emergence of spontaneous structures or self-organized processes in nature: how can a distributed process organize into a collective pattern that maintains some organizational invariance? how does complexity arise spontaneously without an organizing centre? under which structural and environmental or boundary conditions is that possible? One can pose these very same questions in the realm of social life. In fact our social life is often subsumed under emergent collective patterns. A particularly interesting domain to put these type of questions to practical use is the

realm of political organization and grassroots activism in social media. When is a process genuinely self-organized and participatory and when is it just an amplification of an organizing centre of power? Can “genuine” self-organization and the generation of a social “consciousness” be quantified?

The last years have witnessed an explosion of political activism based-on or catalysed-by social media. The Icelandic ‘Kitchenware Revolution’, Wikileaks ‘Cablegate’ and Anonymous’ network defence, the 2011 Arab Spring, the Spanish 15M movement, the Occupy Movement... are but a few among the many examples of the increasing role played by social media in grassroots political organizing. While these and similar social movements differ in many important ways, there is one thing they share in common: they are all interwoven through autonomous communication networks supported by the Internet and wireless communication (Castells, 2012). Social media has provided the tools for creating horizontal and interactive communication networks, boosting enormously the possibilities for self-organized political processes.

The objective of this work is to better analyse and understand social self-organizing processes in the context of social media political activism. More specifically, we centre our analysis in the presence of fractal scaling in the form of  $1/f$  noise in different activist communication processes related to the Spanish 15M movement. We will start describing the context of the Spanish 15M movement and providing a short introduction to the theoretical tools used for our analysis. Next we describe the data, measures and results of the study, and we finally discuss the consequences and limitations of our work.

## 15M Movement and Social Media Channelled Self-Organization

The presence of Internet and other digital media and the increasing use of multidirectional and interactive mass communication networks is starting to change radically the way societies organize themselves to constitute counter power or change power relationships with dominant institutions

(Castells, 2009). An important example of this phenomena is the protest movement that was born in Spain in May 15th 2011 (henceforth 15M movement). Initially, what would later become the 15M movement was organized via social media preparing a massive demonstration without much visibility in traditional media. Unexpectedly, this social media work triggered a huge social movement involving between 6.5 and 8 million people (El País, 2011) that has abruptly changed the political life in Spain. Together with other parallel experiences (like the Arab Springs, or the Occupy movement), the 15M has renewed the communication strategies of previous social movements, now seen as obsolete due to new communicative practices adopted by a large segment of the population (Toret, 2012). Frequently, activists involved with the movement have described its organizational practices as dramatically different to their previous political experiences. These statements suggest that the movement cannot be described just as a group of well-defined formal organizations confronting established institutions. Instead, they often take borrowed dynamicist and complex systems metaphors and describe the organized movement as a swarm or an autopoietic network, with distributed and emergent properties (Sanchez Cedillo, 2012), a system governed by organismic cycles in which the activity of the system is coordinated through powerful synergies (Malo and Pérez, 2012) or as a ‘self-organizing climate’ which envelopes society (Fernández-Savater, 2012).

Unlike previous studies of network analysis of the 15M movement and the similar uprisings, the focus of this paper is on characterizing more global aspects of self-organization processes and exploring indicators of the kind of emergent communication patterns. More specifically, we will focus on the constitution of the system as a coherent whole which can maintain a dynamic identity for a period of time. Since this type of self-organization into a coherent dynamic unit is hypothesized to be the core of mental life and neural organization (Van Orden et al., 2003), we want to explore the possible analogy with social life and political consciousness.

### Fractal Scaling and Self-Organization in complex networks

One of the greatest challenges for the understanding of cognitive and social system is finding formalisms to understand how complex activity emerges from processes of multi-scale organization. During the last decade, different authors have proposed methods of fractal analysis as a solid candidate for this task (Dixon et al., 2012).

Fractal scaling is characteristic of critically self-organized systems (Bak et al., 1987). In these systems we can find an interesting mix between stability and instability creating complex structures of the variability of the system’s activity. Thus, processes with fractal scaling present a constant relation between the size of their fluctuations and the scale in which they occur: systematically larger fluctuations for

longer scales and smaller fluctuations for shorter scales. We often describe fractality in a process through its spectral density function  $S(f)$ , which in the case of fractal scaling exhibits the form:

$$S(f) \propto f^{-\beta}$$

where  $f$  stands for the different frequencies in which the activity of the process takes place, and  $\beta$  defines a log-linear relationship between the spectral power content at different scales of  $f$ . The presence of log-linear relationships in the spectral density suggests that the activity of the system is self-organized into a nested temporal structure, in which the different rates of activity of the components of the system are coupled into a coherent macroscopic whole. More specifically,  $\beta$  shows what is the relative influence of each scale in the system. Different values of  $\beta$  describe different relationships between the weight of fast, medium, and slow timescales in the composition of a self-organized system.

The analysis of the fractal coefficients of a system’s activity has been widely used in neuropsychology for characterizing different states of interactivity among the component of a cognitive system, as well as to predict the emergence of new cognitive structures (Dixon et al., 2012).

More concretely, different values of the  $\beta$  parameter allows us to characterize different types of processes:

- White noise ( $\beta = 0$ ) describes fully random fluctuations with no correlations in time (processes with no memory). White noise processes show a strong dependence on short time scale events (scales with higher frequencies). White noise processes display fast changes in their activity but are unable to maintain structured and coherent patterns.
- Brown noise ( $\beta = 2$ ) resembles a diffusion process with no correlation between increments, but with a strong dependencies between the position of one sample and the next, presenting a “memory” of previous events. It shows a strong dependence on long time events, where small frequencies give a much greater contribution to the noise structure than the rest. Brown noise processes are able to maintain stable structural patterns, but they are unable to flexibly modify their activity when fast changes are required.
- Pink noise ( $\beta = 1$ ) describes processes in which an equilibrium is found between the influence of short, medium, and long timescales. It finds an equilibrium between disordered states with high informational content (white noise) and states with strong memory but low informational content (brown noise). Pink noise processes display dynamics which can maintain stable patterns of activity while being able to flexibly regulate their level of activity.

Fractal behavior has been frequently found in biology, psychology and neuroscience during the last decade, and

fractal analysis have been used for characterizing the underlying structure of system producing complex behaviour. In psychology, fractal coefficients have been successfully used for the better understanding of different types of atypical developmental conditions and the prediction of cognitive outcomes (Dixon et al., 2012). Also, deviations from  $\beta = 1$  fractal scaling, either toward white or brown noise, have been found in different health conditions as epilepsy (Ramon et al., 2008), heart failure (Goldberger, 2002), developmental dyslexia (Wijnants et al., 2012b), among many other examples (Wijnants et al., 2012a).

Although there is still controversy about the meaning of fractal scaling and its origin (Van Orden et al., 2005; Diniz et al., 2011), one position in the debate is that fractal scaling is a characteristic manifestation of self-organizing systems, which reflects the balance between independent and interdependent activity of their components. In this sense, deviations from perfect  $\beta = 1$  pink noise would imply unbalancing this equilibrium in favour of either independent or interdependent activity, affecting the ability of a system to behave in a self-organized manner. Recent years have witnessed increasing empirical support for the idea that pink noise may result from the interaction of many ongoing processes over a multiplicity of interdependent scales (Kello et al., 2007; Dotov et al., 2010; Wijnants et al., 2009).

According to these ideas, the measured  $\beta$  coefficient can account for the type of underlying structures of complex phenomena. The absence of long-range correlations when  $\beta = 0$  implies that the different processes composing the system are highly independent, provoking uncorrelated randomness in the systems activity. This describes a situation where the ongoing activity is not self-organized at all and there is not a coherent collective pattern emerging from their activity. On the other hand we have  $\beta = 2$  processes where slow timescales dominate over others. These systems will display highly predictable patterns which will strongly constrain the individual dynamics of the system components, strangling the self-organization in favour of rigid and inflexible collective dynamics. Finally, when  $\beta = 1$ , slow and fast timescales are compensated, and the influence of independent and interdependent activity is perfectly balanced; the activity of the system is going to depend on the ongoing reciprocal interactions between timescales, finding an equilibrium between stability and spontaneity. As a result  $\beta = 1$  becomes an indicator of distributed self-organization in a coherent whole: different parts of the system (with their characteristic frequencies) appear globally coordinated in a reciprocally influencing manner.

## Methods

Now that we have proposed an indicator for characterizing different self-organization processes, we wish to apply it to answer questions about different instances of grassroots political mobilization. What is the relationship between indi-

vidual political activity and the collective global process? Are processes of political organization of the digital age spontaneous angry mobs or just mindless followers of popular topics? What is the degree of collective 'political consciousness' in social media-based mobilizations?

We now proceed to present different sets of data about a series of political events that we have classified according to qualitative experience from observation and a set some measures aiming to provide quantitative indicators for answering some of the questions above.

## Data and qualitative analysis

We have collected Twitter data from different protests taking place during May 2012, one year after the start of the 15M movement. That month was chosen because of the high density of mobilizations, allowing us to compare different organizational processes taking place in the same short period (thus neutralizing the influence of contextual factors and differences due to large scale variations on composition and methods of the 15M movement through its historical evolution). We downloaded around 385,000 Twitter messages using 20 different *hashtags* (labels used in twitter to identify conversational topics). Following the advice of activists involved within the protests, hashtags were chosen as representative of different types of processes for mobilization and protest:

- Events related to an education strike taking place on May 22 (#22M, #HuelgaDeClase), which follows a more traditional pattern of organization, with unions and other centralized organizations leading protesters, a fixed schedule of pickets, events, demonstrations, and a more predictable process of escalation of the mobilization as the strike date approaches.
- A series of events coinciding with the anniversary of the start of the 15M movement (with the global label #12M15M), from May 12 to May 15, consisting mainly on a series of previously planned demonstrations named through different labels (#ALaPlaza12M, #Es15M, #YoVoy12M, #Feliz15M). Some of these pre-planned events evolved into a series of spontaneous and more creative actions, as a campaign against the Bank 'La Caixa' (#LaCaixaEsMordor) that spontaneously turned into a camp in front of the Headquarters of La Caixa with daily casserole protests under the label (#OccupyMordor).
- Some events correspond to planned actions or proposals launched by some participants of the movement and amplified by the rest of the network in a distributed and decentralized way. This has been a characteristic mode of functioning of the 15M movement, which avoids formal leadership and centralized organization in favour of a more diffuse organization exploiting the possibilities of social media. Some examples of this

are #PlandeRescateCiudadano, #BankiaEsNuestra, #15MSectorRadical, #NurembergFinanciero or #CierraBankia.

- One special case of the category above is the case of #15MPaRato. A team of activists and lawyers launched a campaign aiming to file a lawsuit demanding that the former director of *Bankia* (an Spanish bank accused of being a major responsible for the Spanish financial crisis), Rodrigo Rato and the rest of the director's board, be held accountable for mismanagement and possible criminal behaviour. The campaign was cooperatively designed to synchronously seed the message through a group of well positioned nodes within the movement's network, aiming to work as "catalysts" to create a supporting community for the organization of a *citizen's lawsuit*. This strategy turned out to be very successful, the hashtag #15MPaRato quickly became trending topic in Spain and it reached the first position between the global trending topics soon after, it also managed to collect € 15000 of funding in less than 24h and contacting with tens of Bankia's stockholders and employees willing to take part as witnesses for the trial.
- A last set of events correspond to fast and spontaneous reactions to unexpected incidents, such as the moment where the Spanish risk premium reached the symbolic value of 500 points (#Prima500), the police eviction of a social centre that was central to the movement (#LaRimaia), the eviction of the protesters camped in Sol square in Madrid during the #12M15M protests (#DesalojoSol), or the spontaneous demonstration that followed as reaction to this last eviction targeting Madrid's Stock Exchange Building (#ALaBolsa).

The task is now to test whether fractal scaling analysis can provide a good *quantitative* index for these *qualitatively* different degrees of self organization of social communication and coordination expressed through Twitter.

### Detrended Fluctuation Analysis

The detrended fluctuation analysis (DFA; Peng et al., 2000) is a method for determining the statistical self-affinity of a signal. In a nutshell, the DFA algorithm integrates the analysed time series and then divides it into boxes of equal length  $n$ . For each box and each value of  $n$ , a least squares line (the trend of the signal within that box) is fit to the data. For each box size  $n$ , the characteristic size of the fluctuation  $F(n)$  is computed as the rms deviation between the integrated signal and its trend in each box. This computation is repeated for every value of  $n$ . Typically,  $F(n)$  increases with  $n$ . A linear relationship on a log-log plot with slope  $\alpha$  indicates the presence of fractal scaling in the analysed signal (Figure 1). Where  $\alpha$  is an approximation of the Hurst exponent, and is related to the scaling in the Power Spectrum of the Fourier analysis being  $\beta = 2 \cdot \alpha - 1$ .

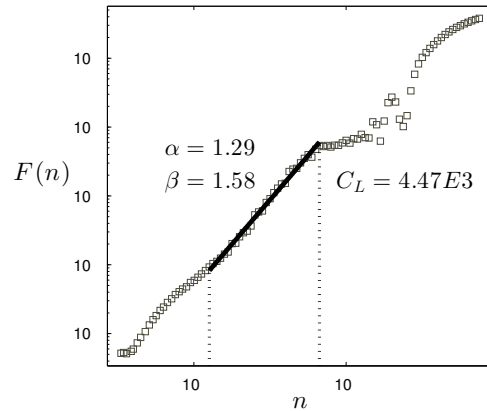


Figure 1: Detrended fluctuation analysis of the time series of tweets with the hashtag #22M. The squares represent the values of  $F(n)$  for each value of  $n$ .  $C_L$  is the larger value of  $n$  where the linear correlation still holds. The solid line represents the linear approximation of the 30 first samples under  $C_L$ .  $\alpha$  represents the slope of the linear fit and  $\beta$  its corresponding value of fractal scaling in the frequency spectrum. Copyright 2013 Miguel Aguilera Creative Commons Attribution-ShareAlike 3.0 Unported License

their common fractal dimension  $FD$ , which for spectral analysis takes the form  $FD = 0.1\beta^2 + 0.4\beta + 1.5$ , and for DFA  $FD = 0.4\alpha^2 - 1.2\alpha + 2$ .

For each hashtag, we created a time series composed of the number of messages that were written using that label at each instant of time (with a sampling period of 1s). The DFA algorithm was applied to the resulting series, obtaining the  $F(n)$  values for each box size  $n$ . After that, we identified the larger value of  $n$  where the shape of  $F(n)$  started to have a log-linear relationship. This value, which we will define as the correlation length of the process ( $C_L$ ) (Figure 1), represents what is the larger scale where the fractal scaling holds. Over larger scales, the fractal relationships break down. We interpret this value as an indicator of how long a process can exist as a self-organized entity. Even if the process exists (i.e. the hashtag is used) for a longer time period, the emergence of coherent patterns of behaviour will not be larger than the value of  $C_L$ . Once identified, this temporal boundary of the self-organized process, we proceed to compute the fractal relations for smaller values of  $n$ . In order to avoid artefacts for small values of  $n$ , potentially due to the chosen sample rate (see Wijnants et al., 2013), we computed the value of  $\alpha$  by fitting a least squares line of the 30 samples from  $C_L - 29$  to  $C_L$  (Figure 1). The obtained trend was transformed into its corresponding value of  $\beta$ .

### Network Analysis

After analysing the scale invariance of the different processes, we have completed our study by analysing the un-



derlying networks generating those dynamics. We have taken the set of interactions between Twitter users (mentions, replies, retweets) using each of the hashtags to create a directed graph where each user is a node and interactions are represented by weighted edges. This graph represents the structure of interactions behind the communicational processes of each hashtag. For each network we have measured the following parameters and properties:

**Degree of a node** The degree of a node  $k_i$  represents the number of incoming and outgoing connections of that node.

**Clustering Coefficient** It measures the connection density among the direct neighbours of a node. If we define  $M_i$  as the number of connections between the neighbours of a node  $i$  we can compute the mean clustering coefficient of a network with  $N$  nodes as:

$$C = \frac{1}{N} \sum_{i=1}^N \frac{M_i}{k_i(k_i - 1)}$$

A high Clustering coefficient implies a robust structure of the network.

**Mean distance** Distance  $d(i, j)$  in a network measures the shortest path between two nodes  $i$  and  $j$ . The mean distance of a network with  $N$  nodes is computed as:

$$L = \frac{1}{N(N-1)} \sum_{i,j} d(i, j)$$

**Small World Networks** Small-world networks are defined as networks with high clustering coefficient and low mean distance (Watts and Strogatz, 1998). These networks have interesting properties which allow to easily find the shortest path between two nodes. Communication processes over small world structures will be much more robust and efficient. We can know if a network has small world properties from its clustering coefficient and its mean distance. Since the clustering coefficient and mean distance are strongly influenced by the size of the network, we have normalized both  $C$  and  $L$  by dividing them by the clustering coefficient and mean distance of a Erdős-Rényi random network with the same size and edge density  $C_r$  and  $L_r$ . A network will have small world properties when  $C_L \gg 1$  and  $L_r \sim 1$ .

**Scale-Free Networks** Small-world networks in nature often are found in the form of scale-free networks. Scale-free networks (Barabási and Bonabeau, 2003) can naturally grow by preferential attachment of its nodes, resulting in networks in which some nodes called ‘hubs’ have a very high connectivity while the majority of the other nodes are poorly connected. Scale-free networks are characterized by a power law in its distribution of node degrees.

$$P(k) \sim c \cdot k^{-\gamma}$$

where  $k$  and  $\gamma$  are constants. All the networks analysed in this work displayed power law distributions of their degree distribution. Thus, we have computed the values of  $\gamma$  for both the incoming and outgoing degree of connections using a linear regression on the logarithmic distribution of  $P(k)$ , obtaining two coefficients  $\gamma_{in}$  and  $\gamma_{out}$  for each network. The  $\gamma_{in}$  coefficient will represent the inequality in how the nodes act as sources of information. A high value of  $\gamma_{in}$  will imply that a few nodes are generating most of the information travelling through the network. In turn,  $\gamma_{out}$  will represent the inequality in how the nodes propagate this information. A high value of  $\gamma_{out}$  will mean that only a few nodes act as amplifiers of information in the network.

## Results

Following the methods above, we have analysed the fractal exponent and the network properties for the 20 hashtags described above.

### Fractal Scaling

In figure 2 we present the obtained values of  $\beta$  and  $C_L$ . As we can see, the values of  $\beta$  range between pink and brown noise (between 0 and 2), with some values very close to pink noise ( $\beta = 1$ ). The results of  $C_L$  also present quite different results, displaying more than two orders of magnitude between the shortest and larger temporal scope of the self-organized coordination.

In a closer inspection of the results, we can observe how different types of mobilization processes can be identified with different values of  $\beta$ . For example, the more rigidly organized process of the education strike (#22M, #HuelgaDeClase) displays values of  $\beta$  closer to the smoother dynamics of brown noise. On the other hand, most of the “spontaneous” processes (#ALaBolsa, #LaRimaia, #DesalojoSol), together with some of the messages amplified by the network (#15MSectorRadical, #NurembergFinanciero, #LaCaixaEsMordor, #BankiaEsNuestra). Finally, some processes seem to achieve an equilibrium between independent and interdependent dynamics and are quite close to pink noise (#15MpaRato, #12M15M and some of its related hashtags, #PlandeRescateCiudadano, #16M, #Prima500, #OccupyMordor), suggesting that these processes reach some middle point between the spontaneousness of white noise and the stability of brown noise.

Figure 2 also shows a correlation between the values of  $\beta$  and  $C_L$ . We have fitted the obtained values of  $\beta$  and  $C_L$  with a rational polynomial function of order two for both the numerator and denominator. The quality of the fit was measured by a  $R^2$  coefficient of 0.69. We observe how values of  $\beta$  closer to one display much higher values of  $C_L$ , suggesting that when the process of self-organization reaches a dynamic equilibrium between independent and interdependent dynamics it spans into much larger temporal timescales.

That is, independently of the real duration of the communication process, pink noise processes present correlations that reach much further in time, maintaining a dynamical coherence that lasts up to days in the cases with highest values of  $C_L$ .

### Network Properties

Once the different types of self-organization processes have been classified according to their fractal exponents, we proceed to compare these results with the properties of their underlying networks.

**Small-World Properties** We have approximated the least squares second order curve of the normalized clustering coefficients and mean distances in relation to the obtained fractal coefficients  $\beta$  (Figure 3). The results show that different types dynamics are related to different types of underlying network structures:

- White noise (low  $\beta$ ): these processes have low values of  $C/C_r$  and  $L/L_r$ , which means that their underlying networks have lower clustering coefficients (less robust structure) and shorter mean distances (faster information transfer). These results coincide with the idea of white noise processes as being spontaneous and viral but poorly organized processes which disintegrate quickly. Low clustering values and short distances also suggest that independent activity of the nodes has a higher influence on the dynamics of the system, since the activity of individual nodes travels fast and there is not a robust structure of stable communities.
- Brown noise (high  $\beta$ ): these processes in turn present higher values of  $C/C_r$  and  $L/L_r$ , meaning that their underlying networks are robust and well structured but have long communication paths. This is also compatible with the idea of brown noise processes being more robust but less flexible. A high clustering coefficient and long mean distance also suggests a bigger influence of interdependent activity, since communities are strong and the transmission of information through the network is slow.
- Pink noise ( $\beta \sim 1$ ): these processes find an equilibrium in which they have quite high values of  $C/C_r$  and not very long mean distances  $L/L_r$ . Again, pink noise processes manage to have the best from white and brown noise processes, being robust at the same time as they are fast in the propagation of information.

**Scale-Free properties** We have approximated the least squares second order curve of the scale-free  $\gamma_{in}$  and  $\gamma_{out}$  coefficients in relation to  $\beta$  (Figure 4). We observe how there is a dependence between  $\gamma_{in}$  and  $\beta$ , finding that white noise processes present a more egalitarian distribution between the nodes that generate the contents of the communication,

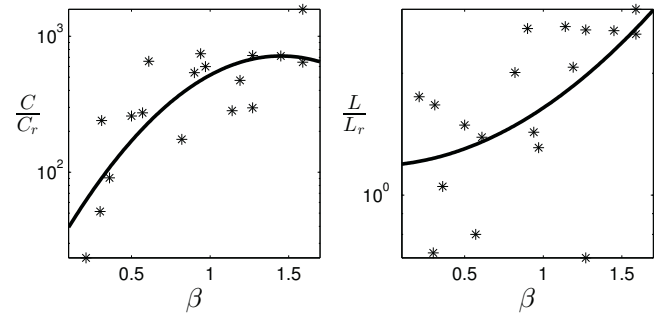


Figure 3: Values of the normalized clustering coefficient  $C/C_r$  and mean distance  $L/L_r$  with respect to the fractal coefficient  $\beta$ . The solid line represents a least squares second order fitting of the data. Copyright 2013 Ignacio Morer Creative Commons Attribution-ShareAlike 3.0 Unported License

while in brown noise processes there are fewer nodes leading the communication process. In the case of  $\gamma_{out}$  it does not seem to have any strong correlation with  $\beta$ , suggesting that the role of the nodes as diffusers of the information is the same independently of the type of the self-organization process going on.

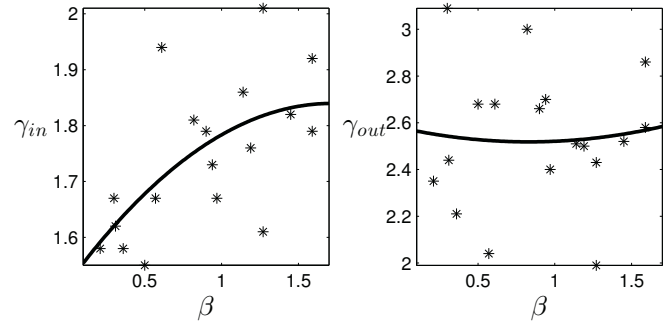


Figure 4: Values of the input and output scale-free coefficients  $\gamma_{in}$  and  $\gamma_{out}$  respect to the fractal coefficient  $\beta$ . The solid line represents a least squares second order fitting of the data. Copyright 2013 Ignacio Morer Creative Commons Attribution-ShareAlike 3.0 Unported License

### Discussion

In previous sections we have presented fractal analysis as a candidate for quantifying and identifying different types of social coordination in communication networks. Concretely, we have proposed fractal scaling in the frequency spectrum of the activity of a connected crowd as an indicator of how it constitutes itself as a collective social system through ongoing interaction. We have measured fractal scaling as the relation between the amount of variability of the system

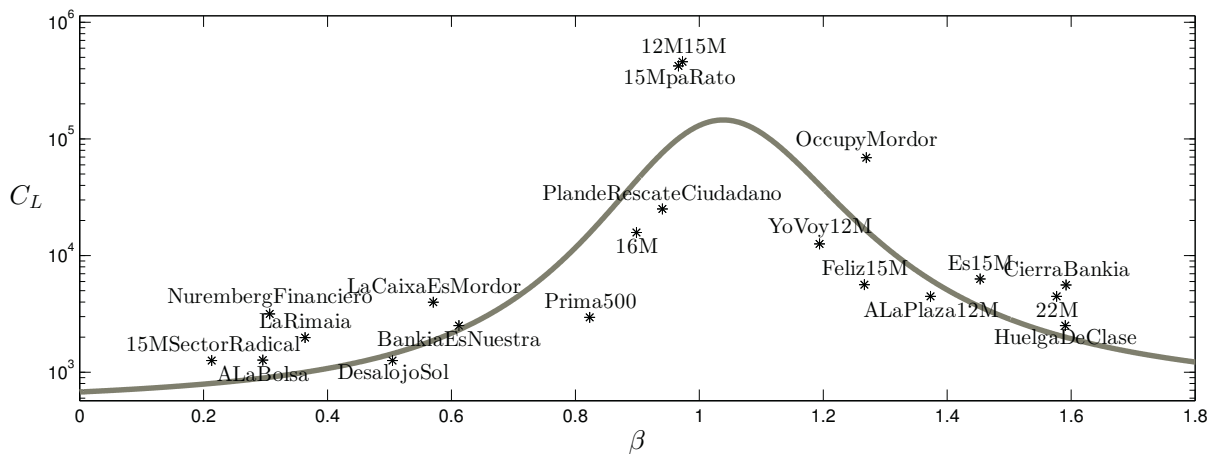


Figure 2: Values obtained for  $\beta$  and  $C_L$  (the latter measured in seconds) for the time series of messages with each hashtag. The solid line represents a rational polynomial fitting of the displayed values. Copyright 2013 Miguel Aguilera Creative Commons Attribution-ShareAlike 3.0 Unported License

at different temporal scales, obtaining a parameter  $\beta$  that describes the fractal relations between the amount of activity in the system at different temporal scales. We have also measured other parameters like the temporal scope of the fractal scaling and properties of the network underlying the communicative activity.

Results have shown how processes which equilibrate the influence of the different temporal scales of activity (those with  $\beta = 1$ ) display properties that suggest that the ongoing process of self-organization is stronger than in other cases, as a larger fractal correlation length  $C_L$  or more marked small world properties. In addition, the processes with  $\beta \sim 1$  coincide with those social mobilizations having a mix of planned or stable development with more spontaneous or surprising turnarounds.

In contrast, processes with an unbalanced dynamical scaling, favouring either short-scale activity or long scale activity, do not display desirable properties for collective self-organization. The larger presence of fast or slow timescales implies that the system is giving preference to either independent or interdependent activity over the other. For example, we have seen how spontaneous reactions of social media activists to unexpected events tend to show white noise scaling, suggesting that the system is not really self-organized into a coherent unit of activity, but is rather the sum of the activities of a uncoordinated crowd reactively triggered by an external stimulus. On the other hand, we have seen how processes organized according to more rigid schemes, like as a strike, tend to brown noise scaling, suggesting that interdependent activity dominates the dynamics of the process, leaving no room for true self-organization as the individual dynamics are enslaved by the collective communicational process.

Despite the consistency of the present results, fractal scaling analysis must be taken carefully. Although fractal scaling is usually taken to emerge as a result of a self-organized processes, there is evidence suggesting that this is not always the case, and there are cases where fractal scaling can be the product of linear combinations processes without fractal scaling (e.g. with a linear superposition of random components acting on multiple time scales, see Hausdorff and Peng, 1996). To avoid this problem, some authors have suggested multifractal analysis to ensure the nonlinear nature of the ongoing interactions that build the self-organized process (Ihlen and Vereijken, 2010). In an extension of the present work (to be published) we have analysed the multifractal structure of some of the data presented here, confirming the relation between pink noise exponents and non-linear self-organized interactions we have claimed here.

Further progress in this direction might demand a more synthetic approach to test how artificial network topologies and communication dynamics yield different forms of noise and correlation length. Artificial Life techniques, like genetic algorithms, could be helpful to find optimal communication strategies in order to build genuine self-organized processes, including parametric analysis of the effect of variables such as degree of consensus, viral potential, external mass-media coverage, etc. Fractal scaling indexes could be used as fitness functions for these models.

## Conclusion

The generalization of social media in everyday communication is a game changer for the analysis and understanding of self-organization in our social and political life. On the one hand, the use of digital communication networks has allowed to overcome some important limitations and diffi-

culties for organizing large scale groups of people without a hierarchical structure (the series of political uprisings we have witnessed in the last 3 years is a good indicator of this potential). On the other hand, social media allows us to easily collect data from social interactions in different contexts. What is still missing is a deeper understanding of how social media network topology and dynamics correlates with different forms of collective action and self-organized coordination.

We have performed an analysis and comparison of 20 different communicative processes related to grassroots political mobilizations within the Spanish 15M movement in May 2012. A qualitative classification of these processes was shown to match the quantitative measurement of fractal scaling analysis of the message exchange time series. A balance between fast and slow temporal scales, described by a *pink noise distribution*, seems to boost the robustness and the life-span of genuine self-organized processes. Pink noise processes in social networks were also shown to be closely related with the stronger small-world properties of their underlying networks. Further analysis and modelling of social and political self-organization is required to support the hypothesis advanced in this paper but we might be witnessing the emergence of a pink noise revolution where it is the dynamics of social interaction (rather than the specific content being communicated) what matters. To say it paraphrasing Marshall McLuhan's celebrated motto, we might have entered an era where *the noise is the message*. And a science of Artificial (social) Life is perfectly suited to push this message forward.

## Acknowledgements

M.A. and M.G.B. are supported in part by the project TIN2011-24660 funded by the Spanish "Ministerio de Ciencia e Innovación". M.A. currently holds a FPU predoctoral fellowship from the Spanish "Ministerio de Educación". X.E.B. acknowledges funding from the research project "Autonomía y Niveles de Organización" financed by the Spanish Government (ref. FFI2011-25665) and IAS-Research group funding IT590-13 from the Basque Government.

Specially, we would like to acknowledge our colleagues of DataAnalysis15M for their work and support.

This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License

## References

- Bak, P., Tang, C., and Wiesenfeld, K. (1987). Self-organized criticality: An explanation of the  $1/f$  noise. *Phys. Rev. Lett.*, 59(4):381–384.
- Barabási, A. and Bonabeau, E. (2003). Scale-free networks. *Sci Am*, 288(5):60–69.
- Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, U.S.A.
- Castells, M. (2009). *Communication power*. Oxford University Press.
- Castells, M. (2012). *Networks of Outrage and Hope: Social Movements in the Internet Age*. Wiley.
- Diniz, A., Wijnants, M. L., Torre, K., Barreiros, J., Crato, N., Bosman, A. M. T., Hasselman, F., Cox, R. F. A., Van Orden, G. C., and Delignieres, D. (2011). Contemporary theories of  $1/f$  noise in motor control. *Hum Mov Sci*, 30(5):889–905. PMID: 21196059.
- Dixon, J. A., Holden, J. G., Mirman, D., and Stephen, D. G. (2012). Multifractal dynamics in the emergence of cognitive structure. *Top Cogn Sci*, 4(1):5162.
- Dotov, D. G., Nie, L., and Chemero, A. (2010). A demonstration of the transition from ready-to-hand to unready-to-hand. *PLoS ONE*, 5(3):e9433.
- El País, E. (2011). Hasta 8,5 millones de españoles apoyan el movimiento 15-m.
- Fernández-Savater, A. (2012). Cómo se organiza un clima? <http://blogs.publico.es/fueradelugar/1438/como-se-organiza-un-clima>.
- Goldberger, A. L. (2002). Fractal dynamics in physiology: Alterations with disease and aging. *Proc Natl Acad Sci*, 99(9001):2466–2472.
- Hausdorff, J. and Peng, C.-K. (1996). Multiscaled randomness: A possible source of  $1/f$  noise in biology. *Phys Rev E*, 54(2):2154–2157.
- Hemelryk, C. K. and Kunz, H. (2003). Introduction to special issue on collective effects of human behavior. *Artif Life*, 9(4):339341.
- Holland, O. and Melhuish, C. (1999). Stigmergy, self-organization, and sorting in collective robotics. *Artif Life*, 5(2):173–202.
- Ihlen, E. A. F. and Vereijken, B. (2010). Interaction-dominant dynamics in human cognition: Beyond  $1/f$  fluctuation. *J Exp Psychol Gen*, 139(3):436–463.
- Kauffman, S. A. (1993). *The origins of order*. Oxford University Press US.
- Kello, C. T., Beltz, B. C., Holden, J. G., and Van Orden, G. C. (2007). The emergent coordination of cognitive function. *J Exp Psychol Gen*, 136(4):551–568. PMID: 17999570.
- Lazer, D., Pentland, A., Adamic, L., Aral, S., Barabási, A.-L., Brewer, D., Christakis, N., Contractor, N., Fowler, J., Gutmann, M., Jebara, T., King, G., Macy, M., Roy, D., and Alstyn, M. V. (2009). Computational social science. *Science*, 323(5915):721–723. PMID: 19197046.
- Malo, M. and Pérez, D. (2012). Latidos: el 15m y la revuelta. In *Democracia Distribuida. Miradas de la Universidad Nómada al 15M*.
- Peng, C. K., Hausdorff, J. M., and Goldberger, A. L. (2000). Fractal mechanisms in neuronal control: human heartbeat and gait dynamics in health and disease. In *Self-Organized Biological Dynamics and Nonlinear Control*. Cambridge University Press.
- Ramon, C., Holmes, M. D., Freeman, W. J., McElroy, R., and Rezvanian, E. (2008). Comparative analysis of temporal dynamics of EEG and phase synchronization of EEG to localize epileptic sites from high density scalp EEG interictal recordings. *Conf Proc IEEE Eng Med Biol Soc*, 2008:4548–4550. PMID: 19163727.
- Sanchez Cedillo, R. (2012). El 15M como insurrección del cuerpo-máquina. In *Democracia Distribuida. Miradas de la Universidad Nómada al 15M*.
- Toret, J. (2012). Una mirada tecnopolítica sobre los primeros días del 15M. In *Democracia Distribuida. Miradas de la Universidad Nómada al 15M*.
- Van Orden, G. C., Holden, J. G., and Turvey, M. T. (2003). Self-organization of cognitive performance. *J Exp Psychol Gen*, 132(3):331–350. PMID: 13678372.
- Van Orden, G. C., Holden, J. G., and Turvey, M. T. (2005). Human cognition and  $1/f$  scaling. *J Exp Psychol Gen*, 134(1):117–123. PMID: 15702967.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature*, 393(6684):440–442.
- Wijnants, M. L., Bosman, A. M. T., Hasselman, F., Cox, R. F. A., and Van Orden, G. C. (2009).  $1/f$  scaling in movement time changes with practice in precision aiming. *Nonlinear Dynamics Psychol Life Sci*, 13(1):79–98. PMID: 19061546.
- Wijnants, M. L., Cox, R. F. A., Hasselman, F., Bosman, A. M. T., and Orden, G. V. (2013). Does sample rate introduce an artifact in spectral analysis of continuous processes? *Front. Physio*, 3:495.
- Wijnants, M. L., Cox, R. F. A., Hasselman, F., Bosman, A. M. T., and Van Orden, G. (2012a). A trade-off study revealing nested timescales of constraint. *Front Physiol*, 3:116. PMID: 22654760.
- Wijnants, M. L., Hasselman, F., Cox, R. F. A., Bosman, A. M. T., and Van Orden, G. (2012b). An interaction-dominant perspective on reading fluency and dyslexia. *Ann Dyslexia*, 62(2):100–119. PMID: 22460607.



# Biology of Digital Organisms: How Language and Tools Construct Reality

Orly (Kramash) Stettiner<sup>1</sup>

<sup>1</sup>Program in Science, Technology & Society (STS), Bar-Ilan University, Israel  
orlyst@netvision.net.il

## Abstract

When Christopher Langton first coined the term "artificial life" and organized the first conference of the nascent field in 1989, he envisioned that "We would like to build models that are so life-like that they cease to become models of life and become examples of life themselves." (Langton 1989). When Thomas Ray referred to his Tierra creatures four years later, he said: "These are not models of life, but independent instances of life" (Ray 1993).

Katherine Hayles, the American postmodern literary critic was startled by this vision and wondered how it was possible, in the late twentieth century, to "believe, or at least claim to believe, that computer codes are alive? And not only alive, but natural?" (Hayles 1996). The American philosopher of science Evelyn Fox Keller supported Hayles's view and generalized it into the linguistic domain (Keller 2002).

In this paper we briefly describe Hayles and Fox-Keller's claims, which will follow by an extended examination of how the usage of language, visualization and analysis tools have continued to construct and shape the field of ALife in the decade since their articles were published. Through this inspection, we suggest that the extensive usage of biological terminology and tools may give researchers a false impression regarding the validity and scientific significance of the experiments involving artificial simulated "organisms".

## Introduction

### Hayles's Narratives

Katherine Hayles concentrated on stories told about, and through, the evolving computer programs, where she identifies two levels of narratives, multilayered systems of metaphors: The textual-visual level and the strategic level.

The first level refers to the textual and visual representations of the environments, in which the artificial creatures develop. "In these representations", she said, "authorial intention, anthropomorphic interpretation, and the program's operations are so interwoven that it is impossible to separate them". Biomorphing naming such as "birth/death", "mother/daughter cell", "ancestor", "parasite", and their redefined interpretations reveal an intention of enabling a dynamic emergence of evolutionary processes within the computerized environment. The visual depiction of the code, in the form of sequenced images, imposes a feeling of real existence of living creatures within the computer. Ignoring the fact that the code is actually the organism and vice versa, the "creatures" gain a phenotypic

expression, a "body"- both visually (through specific shape, size and color) and verbally.

The Strategic level includes arguments and "political" strategies, used to position ALife as a research field within Theoretical Biology. New possible forms of "life-as-it-could-be" emerge spontaneously and evolve within the computer. Such attempts to synthesize life-like behaviors from simple rules and building blocks are claimed to complement the traditional analytic biological sciences that deal with "life-as-we-know-it". The essence of life, narrowed into complex logical forms, is claimed to be independent of medium. Thus, the ALife programs, considered alive themselves, are worth of studying as alternative evolutionary silicon-based pathways, which becomes a model for understanding the natural processes.

These narratives, as Hayles claimed, translate the operations of computer codes into dramatic biological analogues of a Darwinian struggle for survival and reproduction, the rise and fall of races, invented strategies for effective evolution, cooperation and competition.

### Fox Keller's Lexicon

The American philosopher of science Evelyn Fox Keller supported Hayles's view, emphasizing the linguistic domain. Referring to the extensive biological lexicon, which ALife researchers developed for interpreting their models, she wrote, "it adds substantively to the sense of proximity to the real-life examples for which they aim" (Keller 2002, p. 277). Much like Hayles' note that "the organism is the code and the code is the organism", Fox Keller emphasized a persistent ambiguity and even identity of the words "genome", "program" and "body" of digital organisms, as a central agenda.

Fox Keller emphasized the increasingly narrowing and illusory gap between computers and organisms, as reflected by the terms "computational biology" and "biological computation", wondering if this convergence (both material and conceptual) can lead to an indistinguishable gap between the living and non-living. In her analysis, Fox Keller makes two major statements, regarding the achievements, in her view, of ALife simulations at the time of writing (2002).

(1) "The failure (at least to date) to generate the kinds of complex mechanisms observed in biological evolution weakens the claim of such models to enhance our understanding of life-as-we-know-it" (p. 281).

(2) "The models of A-Lifers have thus far failed to engage much interest among their biological colleagues" (p. 283).

At the end of this paper we shall examine the current validity of these two statements.

Over a decade has passed since the prognosis of Hayles and Fox-Keller. The discipline of Artificial Life and digital evolving organisms has become a mainstream, almost fully accepted and established field of research, at least as a method of studying evolution, with its own conferences and publications.

An investigation into some of the leading research reports in this field shows that the narrow trail, which Hayles and Fox Keller identified a decade ago, was followed by others and broadened into an actual highway. To demonstrate this trend we selected to focus on three levels, which we identify as central to the process: The linguistic level, the methodologies and analysis tools level and the human factor.

### The Linguistic Level

At the linguistic level, we join and reinforce Hayles and Fox Keller, through the detection of an extensive and seemingly deliberate massive usage of re-defined biological concepts and anthropomorphisms of digital organisms, creating a vocabulary that becomes the mainstream convention.

The famous Swiss linguist Ferdinand de Saussure (1857-1913), the chief forerunner of structural linguistics, defined language as a collective product of social interactions. A meaning of a word is determined through differences, relative to other words or concepts. These differences structure our perception, and thus- language constructs our perceived reality. Therefore, naming digital objects after salient biological ones (like genotype or mutation) automatically categorizes them in the biological sphere.

During the 1930's, it was the American linguistic Benjamin-Lee Whorf, who advocated the idea that the structure of a language affects the way in which its speakers perceive and conceptualize their world and even their cognitive processes.

In the 1970's, philosophers and other scholars recognized the importance of language as a structuring agent, in what is known as 'the linguistic turn'. This turn began with the post-structuralism movement that followed De-Saussure and included influential theorists, such as Michel Foucault and Jacques Derrida. Language turned from being a tool for communicating messages into the message itself. Through daily usage of language we constitute our reality in an ongoing process of construction, modification and redefinition.

Genetic Algorithms became popular within computer science, with the early studies of Cellular Automata by John Holland (1975). Genetic Algorithms- by their very name and biological origin- naturally used terms adopted from Genetics: A population of randomly generated strings (called chromosomes) constitutes the genotype (or the genome), which encode candidate solutions (called individuals or phenotypes) to an optimization problem. In each generation, the fitness of every individual is evaluated, which serves as a basis for stochastic selection and modification (through recombination and possible random mutations) to form a new evolved population.

During the last decade, this preliminary basic lexicon was warmly adopted and widely extended to describe the digital

organisms, which live and evolve within machines, such as Avida.

The "Avidians", as the citizens of Avida are affectionately called, "can send messages to each other, produce and consume resources, and sense and change their environment's properties". They might be required to "communicate with neighboring organisms" (McKinley, Cheng et al. 2008). They "consume resources and generate by-products that can themselves serve as resources for other individuals" (Yedid, Ofria et al. 2008).

These descriptions and others attribute to these creatures traits and capabilities usually attributed to living creatures, specifically the ability to choose and to make knowledgeable independent decisions. This, in addition to the recurring statement that "Avida does not simulate evolution- it is an instance of evolution" (Pennock 2007), seems like a persistent attempt to designate the digital organisms a status of being an "instance of life" or even "an instance of an intelligent life".

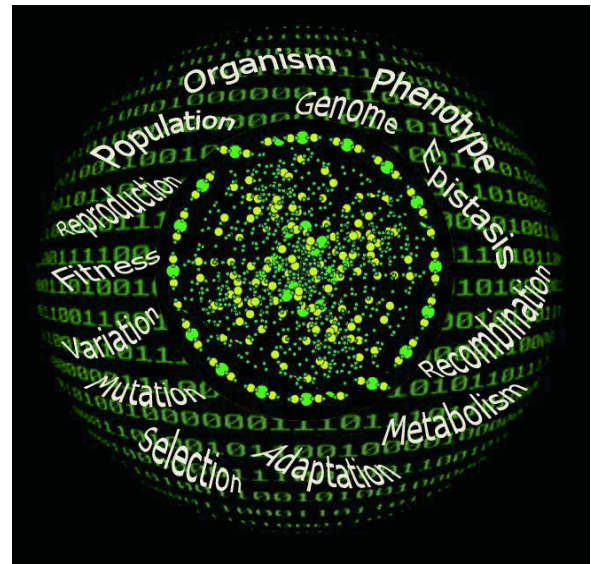


Figure 1: Schematic demonstration of language usage by digital organisms' researchers.

The basic vocabulary created for the original Genetic Algorithms has since expanded vastly to include every aspect of the biological research. It's not just organisms and population, genotype and phenotype, crossover and mutations, birth and death. It now also involves the Metabolism and Metabolic Pathways, Adaptation, Genetic drift and Fixation, Implicit and Explicit mutations, Sexual and a-sexual reproduction, single and multiple niche worlds, and even synergistic and antagonistic Epistasis.

Looking back over three decades ago, one might recall a brilliant critic paper written by an AI researcher, Drew McDermott, who attempted to ridicule some of the common Artificial Intelligence conceptual and linguistics trends, which he calls "mistakes" (McDermott 1976). One such trend was the use of wishful mnemonics: identifiers of programs or data structures named after grand concepts such as "Understand", "Resolve", "Think" or "Associate", describing their desired purpose but not their actual functionality. Referring to the "Is

a..." relation, which is commonly used by AI programmers, McDermott writes:

"Concepts borrowed from human language must shake off a lot of surface-structure dust before they become clear. 'Is' is a complicated word, syntactically obscure. We use it with great facility, but we don't understand it well enough to appeal to it for clarification of anything. If we want to call attention to the 'property inheritance' use, why not just say *INHERITS-INDICATORS*? Then, if we wish, we can prove from a completed system that this captures a large part of what 'is a' means....

People reason circularly about concepts like 'is a'. Even if originally they were fully aware they were just naming *INHERITS-INDICATORS* with a short, friendly mnemonic, they later use the mnemonic to conclude things about 'is a'." (McDermott 1976).

A similar adoption of concepts and naming occurs within ALife, which partly evolved as a separate field in AI, only here the concepts are not borrowed from natural language but rather from Biology. Having recognized the importance of language as a tool for reality construction, we find this usage of biological vocabulary a result of a deliberate affiliation and self-identification on the part of ALife researchers with the discipline of biology, rather than that of computer science. The selected language is becoming a message, it constitutes a modified reality, in which all organisms are alike, the living and the digital, all go through the same evolutionary processes, all become alive, reproduce and finally die, all struggle for survival, where the fittest has the best chances, all compete and sometimes collaborate, they are all actually the same.

## Methodologies, Assessment and Analysis tools

A most fascinating aspect we identified in the process of adopting ALife into mainstream biology is the extensive usage of methodologies, assessment and analysis tools, widely adopted by A-Lifers from molecular biology, evolutionary biology and bioinformatics. We hereby demonstrate five of these tools:

(1) **Fitness Landscapes or Adaptive Landscapes** are concepts created in evolutionary biology, first introduced by Sewall Wright in 1932, to visualize the relationship between genotypes (or phenotypes) and their reproductive success, which is referred to as the "fitness" and visualized by the height at each point of the landscape.

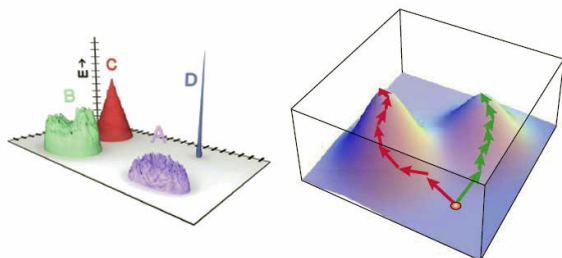


Figure 2: Fitness landscapes of living organisms, right (Elena and Lenski 2003) and of digital organisms, left (Hazen, Griffin et al. 2007)

The concept has gained importance in evolutionary optimization problems, followed by its adoption for describing the fitness function of digital organisms.

In Fig. 2, hypothetical fitness spaces are sketched (on the right) to describe the dynamics of evolutionary adaptation of bacteria and viruses (Elena and Lenski, 2003), while on the left, the fitness function space is shown, describing four classes of possible sequence solutions for digital populations evolving on Avida (Hazen et al., 2007).

(2) **Phylogenetics** is the study of evolutionary relations among biological species or populations, which is usually discovered through molecular sequencing and morphological data. Phylogenetic depth refers to the cumulative number of generations or lineages by which organisms differ from their common ancestor. A usage of this tool was presented, for example, in *Evolutionary Biology* magazine for the analysis of a large protein database, and in *"Systematic Biology"* journal to show the speciation of two groups of beetles

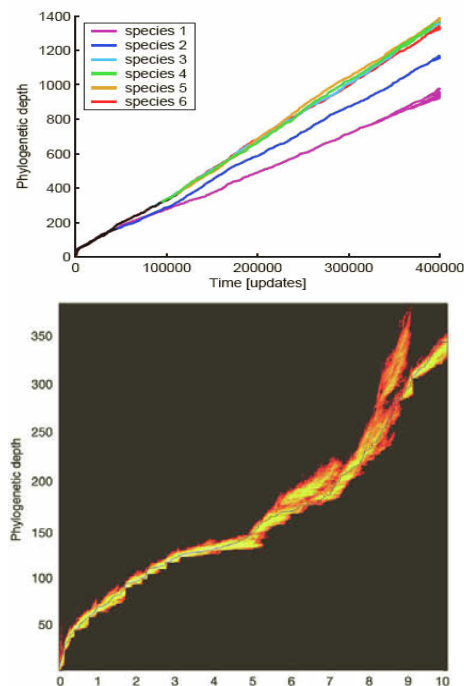


Figure 3: Phylogenetics analysis of digital organisms as presented by (Lenski et al, 2003) on the right and (Chow et al. 2004) on the left.

(*Aphanarthrum* and *Coleobothrus*).

Researchers of digital organisms adopted the concept and tool to present phylogenetic depth analysis, based on the number of generations in which an organism's genotype differs from its parents, where the colors indicate the relative abundance of genotypes at a specific depth (Lenski et al, 2003), (Chow et al. 2004).

(3) **Gene Expression Profiling Analysis** – also known as Functional Genomics Array, is a strategy developed by molecular biologists to describe genes functions and interactions. The profiling can present the measurement of the activity (expression) of thousands of genes at once, to create a global picture of cellular functionality under specific conditions. Each column usually represents a specific



experimental condition, whereas each row stands for a particular gene. A color-coded scale is used, where red generally represents expression greater than a certain reference, green is less than that reference, and gray or white is missing or excluded data (for example, Fig. 4A from (Glaser 2011), Fig. 4B from (Shoemaker et al. 2001).

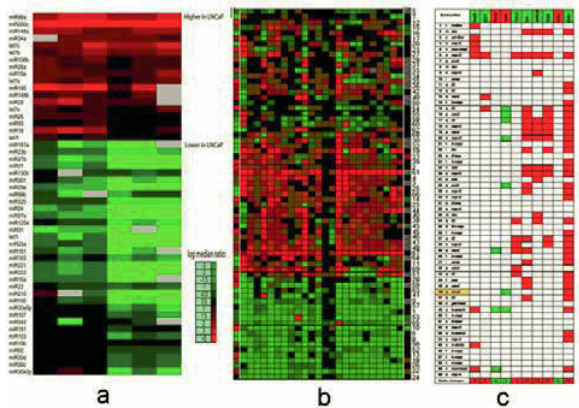


Figure 4: Gene expression profiling as used for living organisms: (a) by (Glaser 2011), (b) by (Shoemaker et al. 2001) and for digital organisms (c) by (Adami 2006).

The strategy was adopted by ALife researchers (e.g. (Adami 2006), as shown in Fig. 4C) to show the effect of knocking out individual computational instructions on the functions (which are actually the Genes) of a digital organism. The color codes look familiar: white indicates an unaffected function, red signifies a turned off (not expressed) function, whereas green signals functions that are turned on (the gene is expressed).

The overall resemblance between the original and adopted tool is remarkable and seems to be deliberate.

(4) **Sequence Alignment** is widely used in bioinformatics to arrange sequences of DNA or RNA (composed of nucleotides) or proteins (composed of amino acids), to identify regions of similarity, reflecting conserved regions, a consequence of functional, structural or evolutionary relationships between the sequences (e.g. (Brachner et al., 2012) (Karpinets et al., 2010), Fig.5A, 5B).

Researchers of digital organisms adopted this technique (Fig.5C, Adami 2000), to visually present the genome sequences of an entire Avida population, existing at a specific generation. The sequencing results demonstrate- at each genome site- the level of entropy, meaning- how variable or conserved this site is. Red sites are highly variable, whereas blue sites are conserved (having low entropy), as can be expected from the common practice in genomic sequencing of living organisms.

(5) **Gel Electrophoresis** is a well known procedure used in molecular biology to separate and sort a mixed population of DNA and RNA fragments by length, or proteins by electric charge, when they are made to move through a gel, usually made of PolyAcrylamide. DNA may be visualized using Ethidium Bromide which, when intercalated into DNA, fluoresce under ultraviolet light.

Recently, an attempt to sort and recognize digital viruses (referred to as "malware" or "cyber organisms") were made, using the same adopted procedure (Jaenisch 2010). The digital viruses, considered to be a collection of polypeptides, forming information-bearing protein structures, were analyzed using a mathematical analog to the 2-Dimensional PolyAcrylamide gel electrophoresis process, where again- the colors and general appearance seem familiar (Fig. 6).

These examples of methodologies and analysis tools, which originated in mainstream fields (such as molecular or

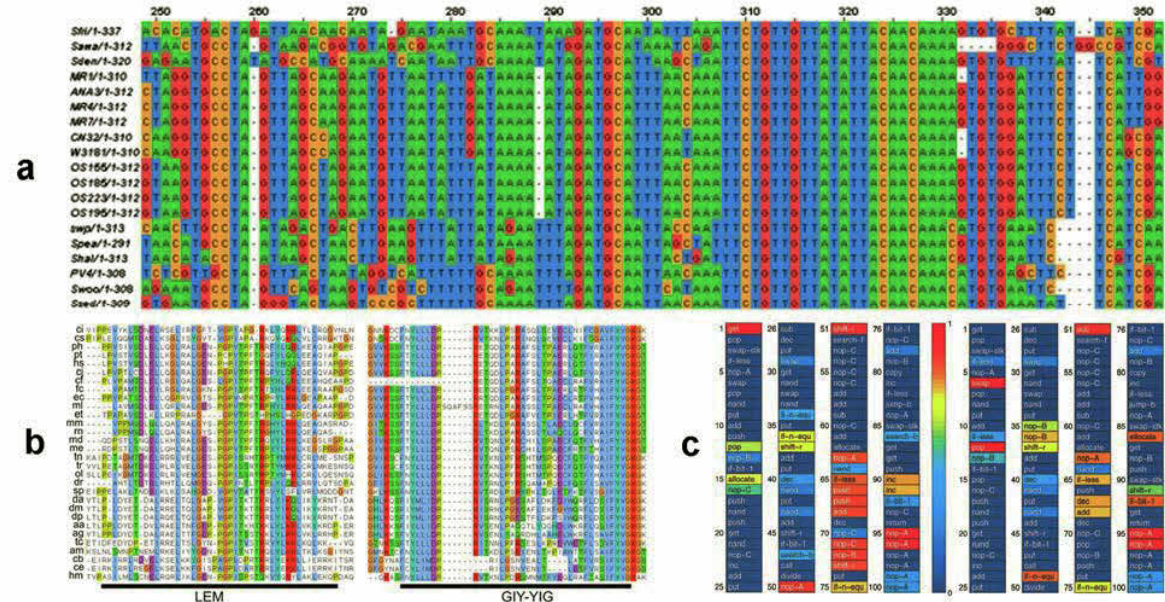


Figure 5: Examples of sequence alignments used for living organisms ((a) From (Karpinets et al., 2010), (b) From (Brachner et al., 2012)), and for digital organisms, (c) from (Adami 2000).



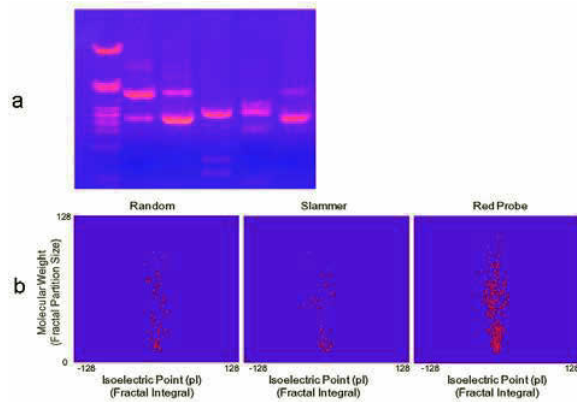


Figure 6: Gel electrophoresis examples as used for (a) proteins, from Wikimedia Commons

[http://commons.wikimedia.org/wiki/File:Gel\\_electrophoresis\\_2.jpg](http://commons.wikimedia.org/wiki/File:Gel_electrophoresis_2.jpg)

and (b) digital viruses (Jaenisch 2010).

evolutionary biology), demonstrate a remarkable visual and conceptual resemblance to the originals. A layman reader of an academic paper or, more importantly, a wet biologist or theoretical evolutionist, seeing the results of a gene expression profile or an electrophoresis slide, might only see the expected familiar visual structure and colors, and will not necessarily pause to make the distinction between the analyzed objects. This obviously results in the intensification of the illusion of similarity and even identity between the living and the artificial species.

### Human factor, publications and citations

Finally, we identify the human factor as having a major influential effect on the increased acceptance of ALife into mainstream biology. The discipline today is composed of a mixture of computer scientists, engineers and leading biologists. Names like Richard Lenski, a distinguished professor of Microbial Ecology, who is well known for his long-term *E. coli* evolution experiments, certainly provide academic credibility, when conducting evolution experiments on other kinds of organisms, either in an actual petri-dish or in a so called "virtual-petri-dish". Christopher Adami, a professor of Microbiology and Molecular Genetics, can easily share the methodologies and assessment tools, used to analyze the genetic traits of Avidians with other Molecular Genetics researchers. As a result, the digital organisms' research has become a mainstream instrument that carries the knowledge and methodologies of related biological fields into the computerized artificial domain. The analogy between *E. coli* and digital organism is easily made, based on their visually comparable circular genome (Fig. 7).

Consequently, we identified an increased reliance of digital evolution studies on core-biology publications and lab reports, and vice versa. Relationships between digital organisms are compared to those reported for long-term bacterial experiments (e.g. (Yedid, Ofria et al. 2008)). ALife papers regularly quote dozens of biological articles, written by Zoologists, evolutionary biologists and molecular biologists,

which- to the reader- gives a definite impression of being part of a much larger corpus of scientific biological research.

Side by side, an increasing number of so-called "pure" biologists conduct experiments on digital organisms, quoting such digital findings and support their real-life conclusions by these results. Digital and Living organisms' based experiments are quoted side-by-side, as reliable sources of theories that explain, for example, the evolution of complex traits. More and more biologists seem to believe that digital organisms' evolutionary mechanism can produce new scientific knowledge and explanation. Such papers are widely found in publications such as: Journal of Evolutionary Biology, Genome Research, Journal of Molecular Evolution, Cellular and Molecular Life Science, and others.

One can read, for example, a quote from a group of Developmental Biology Researchers saying:

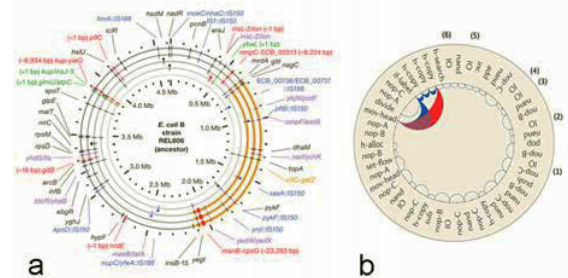


Figure 7: A Scheme of (a) *E. coli* genome (Barrick, et al. 2009) and (b) of and Avidian genome (Adami 2006)

"Although the digital creatures in these experiments embody an abstract view of life... it is highly informative to observe, in a laboratory microcosm, the ability of ever-more fit organisms to emerge while less fit variants disappear from the population..." (McAdams, Srinivasan et al. 2004)

Or a group of biologists conducting Zoological research that writes:

"In so far as the evolution of complex biological structure, under the influence of natural selection, can be expected to follow the same logical rules as these computer simulations [of (Lenski, Ofria et al. 2003). O.S], then the correlated progression model is corroborated." (Kemp 2006).

These developments can easily fix the notion of resemblance and equal academic credibility between the digital computer simulations and in-vivo laboratory experiments of evolving organisms.

Historically, such cross-discipline personal influence can be detected. A well-known example is Erwin Schrödinger's 1944 monograph "*What is Life? The Physical Aspect of the Living Cell*", a book considered one of the most influential scientific books in the twentieth century. Schrödinger, as a Nobel Laureate Physicist had its greatest influence on physicists, who were inspired to emigrate to the field of biology. But, according to the biographer Walter Moore, the book encouraged biologists to think more rigorously, in terms of mathematically formulated and physically testable models, bringing physics to the attention of biologists as well as biology to the attention of physicists (Ceccarelli 2001, p. 63).

## Summary and conclusions

Katherine Hayles said, in her previously mentioned work, that "ALife researchers joke, that ALife is a solution in search for a problem". This is no more.

In 2003, Chris Adami was quoted to have said, when he referred to Avida:

"I wanted this digital life system to be an experimental system just like, let's say, Rich Lenski and *E. coli* bacteria...." (O'Neill, B., 2003).

It seems today that this vision has been fully realized. Having revisited Katherine Hayles and Evelyn Fox-Keller with an updated inspection, we feel strongly that their initial feeling was correct.

In contrast to Fox-Keller's belief, that artificial models will not enhance our understanding of life-as-we-know-it, and her statement that these models "failed to engage much interest among their biological colleagues", we witness biology experts today that quote ALife experiments and treat them as authorized and reliable sources of information on questions that used to be purely in-vivo or in-vitro issues. At the same time, ALifers refer to actual biological dilemma and theories, which they suggest to contribute to, with their powerful and effective computerized tools and through the vivacious, energetic, struggling and reproductive citizens of Avida.

A clear trend of building narratives around digital organisms was identified, which is primarily based on a wide and thorough linguistic biology-based lexicon. Analysis tools and methodologies have been adopted from molecular and evolutionary biology and seamlessly converted into the digital domain, to produce an illusionary feeling that one actually reads and talks about similar entities, with comparable credibility and analogous results. Significant support has been added to this conceptual construction during recent years by leading researchers, who conduct parallel experiments on both living and digital organisms, producing publications that enhance the validity and scientific significance of experiments involving artificial simulated "organisms" and consequently render the narrow gap unnoticeable.

## References

- Adami, C., Ofria, C., & Collier, C. T. (2000). Evolution of biological complexity. *PNAS (Proc. National Academy of Science)*, 97(9), 4463-4468.
- Adami, C. (2006). Digital genetics: unravelling the genetic basis of evolution. *Nature Reviews Genetics*, 7, 109-118.
- Barrick, J. E., et al. (2009). Genome evolution and adaptation in a long-term experiment with *Escherichia coli*. *Nature*, 461(7268), 1243-1247.
- Brachner, A. et al. (2012). The endonuclease Ankle1 requires its LEM and GIY-YIG motifs for DNA cleavage in vivo. *J. Cell Science*, 125:1048-1057. JCS098392 Supplementary Material.
- Ceccarelli, L. (2001). *Shaping Science with Rhetoric: The Cases of Dobzhanski, Schrodinger and Wilson*. Chicago and London, The University of Chicago Press.
- Glaser, V. (2011). Sequencing Alters Medical Genetics. *Genetic Engineering & Biotechnology News*, 31(21):34-36.
- Harris, R. 1988. *Language, Saussure and Wittgenstein: How to play games with words*. Routledge.
- Hayles, K. (1996). Narratives of Artificial Life. Futurenatural: Nature, Science, Culture. M. M. George Robertson, Lisa Tickner, Jon Bird, Barry Curtis and Tim Putnam (editors). London, Routledge: 147-164.
- Hazen, R. M., P. L. Griffin, et al. (2007). Functional information and the emergence of biocomplexity. *Proc. National Academy of Science*, 104(Suppl. 1): 8574-8581.
- Jaenisch, H. (2010). Digital Virus Recognition. *SPIE Newsroom*. DOI 10.1117/2.1201005.002933.
- Karpins, T.V. et al. (2010). Shewanella knowledgebase: integration of the experimental data and computational predictions suggests a biological role for transcription of intergenic regions. *Database*: 2010, Article ID baq012, doi:10.1093/database/baq012.
- Keller, E. F. (2002). *Making Sense of Life: Explaining Biological Development with Models, Metaphors and Machines*. Cambridge, Massachusetts, Harvard University Press.
- Kemp, T. S. (2006). The origin of mammalian endothermy: a paradigm for the evolution of complex biological structure. *Zoological Journal of the Linnean Society*, 147: 473-488.
- Langton, C. G. (1989). Artificial Life- the proceedings of an interdisciplinary workshop on the synthesis and simulation of living systems. *Artificial Life*, Redwood City, CA, Addison-Wesley.
- Lenski, E. R., C. Ofria, et al. (2003). The evolutionary origin of complex features. *Nature*, 423: 139-144.
- McAdams, H. H., B. Srinivasan, et al. (2004). The evolution of genetic regulatory systems in bacteria. *Nat Rev Genet*, 5(3): 169-178.
- McDermott, D. (1976). "Artificial Intelligence Meets Natural Stupidity." *SIGART Newsletter* 57: 4-9.
- McKinley, P., B. H. C. Cheng, et al. (2008). Harnessing Digital Evolution. *IEEE Computer*, 41(1): 54-63.
- O'Neill, B. (2003). Digital Evolution. *PloS Biology*, 1(1): 11-14.
- Shoemaker, D. et al. (2011). Experimental annotation of the human genome using microarray technology. *Nature*, 409: 922-927.

# Evolved Sensitive Periods in Learning

Kai Olav Ellefsen

Department of Computer and Information Science,  
Norwegian University of Science and Technology  
email: kaiolae@idi.ntnu.no

## Abstract

In this paper we study age-varying plasticities across different components in an artificial neural network performing a reinforcement learning task. An evolutionary algorithm is given the task of mapping the age of agents to the plasticity levels of different network components. The results show that patterns of plasticity resembling biological sensitive periods appear, and that these periods schedule learning across the components of the network, which leads to a reduction in the total learning effort while retaining the quality of learning. The sequencing of sensitive periods forms a cascade of partially-overlapping learning periods, which has been proposed as a way of organizing sensory development of abilities that depend on several interrelated brain functions.

## Introduction

Periods in the life of an individual where environmental stimuli are of particular importance for the development of a certain ability are called *critical periods* or *sensitive periods*. They were originally called critical periods to emphasize how lack of the correct stimuli would lead to the sensory system not developing as in normal individuals. In other words, the sensory input is *critical* for neural development. Hubel and Wiesel's classic paper (Hubel and Wiesel (1970)) illustrates this: One eye of a kitten was sutured in various periods throughout life, and it was found that visual deprivation of one eye early in life prevented it from following the regular path of development, leaving the cat blind in that eye, even when it was opened later in life. The period where sutures had this effect was found to have a very specific beginning and end (about four weeks and three months of age, respectively), and this finding is typical of how a critical period was interpreted: A time period with a strict beginning and end, where sensory stimuli have large effects on neural development, and where sensory deprivation leads to abnormal development.

Later research (see for instance Lewis and Maurer (2005)) has shown that sensory systems have *several* critical periods that affect different parts of the sensory system at different stages of development. It has also been shown that these periods can be flexible, and their timing may be controlled by

*experience* rather than age. For example, dark-reared kittens can have the ending of their critical period for vision delayed due to the lack of visual stimuli (Trotter et al. (1981)). These findings have led many researchers to adopt the term *sensitive period*, to emphasize that the period shows a great deal of variation a) across *individuals* that have different experiences during development, b) across *sensory systems* in the same individual and c) across *different parts of a single sensory system*.

An example of the last type of variation was studied by Harwerth et al. (1986). The authors studied the effect of monocular deprivation on different visual functions in rhesus monkeys to investigate the timing of sensitive periods of different functions within a single sensory system. They found there to be *several partially-overlapping* sensitive periods within the visual system, and basic functions (such as spectral sensitivity) were found to have shorter sensitive periods than more complex functions (such as binocular vision). Knudsen (2004) suggests that this property is likely to be found also in other parts of cognitive development, such as in the development of language and social skills. It is logical that low-level behaviors should finish their sensitive period before high-level behaviors, because the low-level outputs will be noisy until these systems have matured, and it will not be possible for the high-level systems to learn from these noisy signals.

Werker and Tees (2006) reviewed findings about sensitive periods in speech processing. The authors argued that speech processing, like vision, depends on a number of inter-related, *hierarchically ordered* brain functions. Based on a review of studies in language development, the authors suggested a possible way for different levels of language learning to be organized through development, where low-level functions once again tend to stabilize before higher-level functions can initiate their sensitive period. Figure 1 shows how they envisioned a *cascade* of sensitive periods in speech processing.

We will mainly use the term *sensitive period* for the remainder of the paper. The term is meant to refer to a period of *heightened plasticity* and sensitivity to environmen-

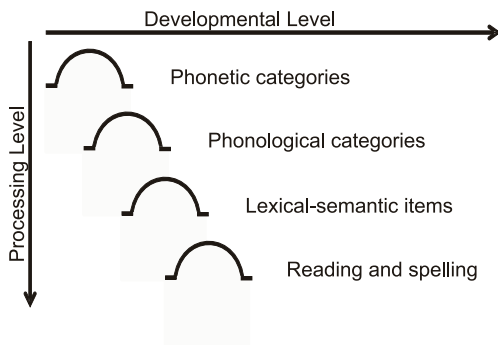


Figure 1: Sensitive periods in speech processing, as suggested by Werker and Tees (2006).

tal stimuli for a given function. If we want to emphasize that the period has a strict ending and that there is no plasticity outside the period, we will use the term *critical period*.

## Background

### Factors behind sensitive periods

Much work has been done in studying which factors *drive* sensitive periods. Armstrong et al. (2006) reviewed some of this work, pointing out two categories of changes that govern their initiation and termination: genetically-mediated changes and experientially-mediated changes. In the former category, sensitive periods will begin and end at a predetermined age as a consequence of innate traits. The latter category concerns factors affecting sensitive periods differently depending on the individual's sensory-input history. Hensch (2005) reviews findings regarding sensitive-period plasticity in the primary visual cortex. He shows evidence from genetically altered mice that the sensitive period in this part of the visual pathway is related to the maturation of *inhibitory circuits*.

Johnson (2005) presents three views on what causes plasticity to decrease as a sensitive period ends: (a) Maturation changes, (b) self-termination of learning, and (c) stabilization of the constraints of plasticity. The first explanation describes chemical processes in the brain that terminate learning independently of experience. The second explanation describes how the learning itself may lead to a decrease in plasticity. One example of a factor leading to self-termination of learning is limited computational resources: when a lot has been learned, there is simply not the same capacity for adding new knowledge. The third explanation regards how sensitive periods may end due to a stabilization in external factors, such as bodily growth, rather than an actual decrease in plasticity.

### Sensitive periods in food preference formation

The experiments in this paper use food-preference formation as the domain for studying sensitive periods. The reason is

that this domain gives us a natural way of splitting the learning task into several subtasks at different levels in a hierarchy, each depending on subtasks below. This is essential if we hope to observe sensitive periods forming in a cascading manner.

Sensitive periods in food preference learning have been studied among animals, for instance snapping turtles (Burghardt and Hess (1966)), lynx spiders (Punzo (2002)) and cuttlefish (Darmaillacq et al. (2006)). Also among human children, a sensitive period of food preference learning has been suggested (Cashdan (1994)).

### Evolving sensitive periods

Bullinaria (2003) studied sensitive periods of learning, as part of a simulation of the human oculomotor system. By the use of an evolutionary algorithm, *age-dependent neural plasticity* was generated. The type of age-dependent plasticity arising from these experiments had parallels with biological sensitive periods. The evolved sensitive periods had the effect of letting individuals be plastic as their sensory systems underwent development, and less plastic after their development was done. Bullinaria describes two simplifying assumptions made in this work that will be important to address in the future. First, plasticities are fully determined by the genotype, meaning *experience* has no effect on the mapping from age to learning rates. Second, the evolved age-dependent plasticity is the same for all parts of the network. In this paper, we remove the second simplification, to study the effect of different evolved learning rates on different parts of an agent's behavior. In a later paper (Bullinaria (2009)) Bullinaria found that a longer period of parental protection gave a lengthening of learning periods in children.

Kirby and Hurford (1997) studied *incremental learning* in language acquisition. By using an evolutionary algorithm to decide the timing of increments to learning capacity, they aimed to study *when* and *why* sensitive periods would form as an effect of incremental learning abilities. They enabled their evolutionary algorithm to shape incremental learning in two ways. It could determine learning resources on the basis of the *age* of an individual or the *experience level* of the individual. Kirby and Hurford found that evolving learning resources based only on the age of an individual gave extreme sensitive periods, similar to what has traditionally been called *critical periods*: A slight delay in the expected stimuli made individuals miss the evolved window for learning, unable to learn language at all. Evolving resources based only on the experience level of an individual gave the opposite effect: No sensitive periods were formed at all – experience could be postponed indefinitely and learning would proceed as normal. It was finally found that letting evolution *combine* the two forms of learning control would give sensitive periods similar to the ones seen in language acquisition in humans.

Hurford (1991) set up an evolutionary algorithm to model



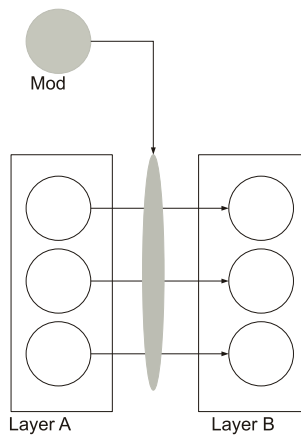


Figure 2: A model of neuromodulated plasticity. The activity of modulatory neurons affects the plasticity of connections between regular neurons.

*language acquisition*. This is a task suggested to have a sensitive period, based on findings from, among other, language recovery in children and adults suffering from aphasia. The evolved learning efforts showed plasticity of the language system peaking in the first period of the individual's life, and gradually falling off to zero. Hurford hypothesizes that the ending of the sensitive period is not happening because it is beneficial to turn off language learning. Rather, as individuals master the language fully at a young age, the pressure to boost language learning is simply gone, so there is no pressure to drive the sensitive period into adulthood. This hypothesis is strengthened by the observations that individuals subjected to a high chance of “language amnesia” at some stage in life tend to evolve lifelong sensitive periods.

### Neuromodulated learning

In the experiments presented here, we study plasticity regulation *across several modules* of a neural network. To exchange reinforcement signals between these modules, we use *neuromodulated learning*.

Neuromodulated neural networks are networks that include another type of signal in addition to the traditional activity-propagating signals. Fellous and Linster (1998) present a review of work on these kinds of networks, revealing that modulatory signals have been used to affect network function in diverse ways, most of them biologically plausible, or at least biologically inspired.

Of particular interest in this context, is the use of neuromodulation to allow efficient reinforcement learning, as that is the role modulation plays in the experiments presented in this paper. In the model of neuromodulated reinforcement learning we employ (Figure 2), modulatory neurons affect the *plasticity* of connections between regular neurons. A single modulatory neuron affects all connections in a *link*, a concept which will be defined in the experimental setup-

section. The model we used is similar to that used in (Soltoggio et al. (2008)), which allowed an agent to learn from sparse events, by letting the modulatory signals have a multiplicative effect on neural plasticity.

### Research Questions

The experiments discussed herein study the emergence of sensitive periods in simple agents in an ALife environment. The environment was set up with a “hierarchical” learning task, where learning of high-level behaviors is dependent on the low-level behaviors already being stable. An evolutionary algorithm (EA) was used to search for optimal sensitive periods for different behaviors. The questions we want to answer are:

- Will the agents show sensitive periods that sequence learning in order of the complexity of behaviors (from low-level to high-level)?
- Will the evolved sensitive periods be able to reduce the agents' learning efforts, while still allowing them to learn the correct behavior?

### Experimental Setup

#### SEVANN

To evolve neural networks with age-varying plasticity, the system SEVANN (Script-Based Evolution of Artificial Neural Networks) (Downing (2010)) was used. SEVANN is a system that lets the user form an *underspecified script* defining an ANN, and then searches for good values for the unspecified parts by use of an evolutionary algorithm. For the experiments reported here, the architecture and initial weights of the ANN were fully specified in the scripts, to allow SEVANN to focus on evolving plasticity values through life, and to allow analyses of results to only depend on the plasticity.

In SEVANN, networks consist of *layers* and *links*: Layers are a number of *neurons* that share common attributes, as well as common inputs and outputs. Links are a number of *arcs* (single connections between neurons) that are grouped together, because they have common attributes, and connect the same layers. When working with age-varying plasticities, we encode learning rates on the link level, allowing evolution to differentiate learning rates between links, but not between single arcs within a link. This granularity was chosen because different links learn different behaviors, and thereby we may allow the sensitive periods to *sequence* behavior learning.

#### Food-Gathering Task

To study how sensitive periods may evolve to sequence the learning of several behaviors, a task with dependencies between different behaviors is needed. The task must have

different levels of behavior, where one level affects performance of the level above - this way, a *cascade* of sensitive periods, as suggested by Werker and Tees (2006), may emerge as a solution to the problem.

Agents are placed on a toroidal grid with food and poison randomly scattered around. To make the hierarchy of behaviors more complex, food is placed in *nuts*, which need to be cracked open to access the food inside. So, the agent has to make three kinds of decisions: where to navigate, which nuts to open, and which food elements to eat. The decisions depend upon each other in a bottom-up fashion: To learn which nuts to open, the agent must first have understood which foods are healthy, then link these to the nuts containing them. And to learn which nuts to navigate to, the agent must first have learned which nuts it wants to open.

Eating food, opening food nuts and visiting food locations increases the agent's fitness, whereas eating poison, opening poison nuts and visiting poison locations decreases it by the same amount. The layout of the food-gathering grid was randomly initialized in each fitness evaluation, with a given probability of each cell containing a poison nut, a food nut or nothing.

An agent resulting from a successful run of the EA typically opens all food nuts and eats all foods. It eats one or a few poison items before learning these should be avoided. Thereafter it still *opens* poison nuts until it learns they are associated with poisons. Next, it passes over poison nuts without opening them, before finally learning to steer away from poison nuts. The order of learning here reflects the hierarchical ordering of the task, and sensitive periods are expected to form in the same order. Each individual was tested *five times* during each fitness evaluation, on grids with different positions of food and poison. This was done to make agents form *general* sensitive periods, instead of sensitive periods adapted to one particular environment.

### Food-Gathering Network

The network structure was chosen to enable learning to propagate from “low-level” (food preferences) to “high-level” (movement preferences) behaviors. The network structure is depicted in Figure 3. It utilizes *neuromodulatory neurons* to be able to transfer what it learned at lower levels to higher levels.

The network mirrors the hierarchical arrangement of the task: Higher levels of behavior are learned by use of reinforcing signals from lower levels. For instance, to learn which nuts to crack (in the link between **InputNut** and **DecisionCrack**), the output from evaluating food is used as a reinforcing signal, with a delay of one timestep: if the food evaluated in the current timestep is good, the decision to crack the nut in the previous timestep was obviously good, and should be repeated in the future. Otherwise, the decision should be avoided, and the arc causing it should weaken.

The bottom layer in each behavior is used to *scale* the out-

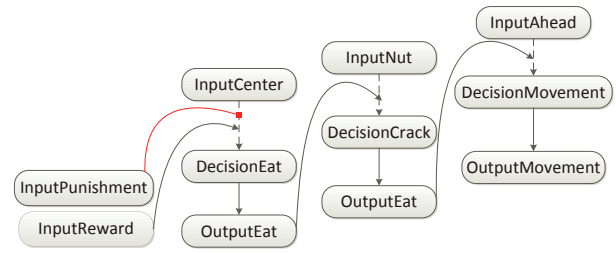


Figure 3: The Neural Network that controls food-gathering agents. Rounded rectangles in the figure correspond to **layers** of neurons. Arrows are **links** between the layers. Dotted lines are **plastic links**. Links targeting other links are **modulatory links**, regulating their plasticity. The link ending in a square signifies a negative modulatory activity, whereas the other modulators give positive reinforcement. The plastic links are all initialized with positive values, to facilitate initial exploration.

put activity of that behavior, so that reinforcing modulation on each level is of the same magnitude. Without this scaling, it would be more difficult to compare learning rates across layers, because layers with a very strong modulatory input could learn with a very low learning rate.

### Learning

Arcs in the network are updated by the following learning rule:

$$\Delta w_{ij} = \eta * mod * |x_i x_j| \quad (1)$$

where  $\eta$  is the learning rate, *mod* is the strength of incoming neuromodulation and  $x_i x_j$  is the product of pre-synaptic and post-synaptic activity, in other words a regular Hebbian update term.

As the equation shows, it is the *absolute value* of the hebbian update that is used in the calculation of the new weight value, since we want the *modulatory signal* to decide the direction of the weight change: negative modulation means whatever action was taken was a bad idea, so the weight of the link causing the action should be decreased. Positive modulation should have the opposite effect. In the absence of modulation (in other words, if  $mod = 0$ ), weights are not updated.

The age-dependent learning rate is encoded as a sequence of real values ( $\vec{\eta}$ ) in the genome. One such sequence is evolved for each plastic link in the network. Each value in the sequence describes the *change* in learning rate for the current age. The first value in the sequence encodes the *initial* learning rate for the link. A separate parameter decides the *delta age*,  $\delta_{age}$ , of the link. This parameter tells the algorithm how often it should update its current learning rate. For instance,  $\delta_{age} = 5$  would mean the rate is updated every

Parameter	Value
Generations	100
Adults	30
Children	50
Crossover probability	0.1
Mutation probability	0.005
Genes per individual	75
Bits per gene	8
Elite fraction	0.2
Culling fraction	0.25

Table 1: Parameters of the Evolutionary Algorithm

fifth timestep, giving a total of 20 updates for a 100-timestep run. A reasonable setting of this parameter allows us to reduce the complexity in finding a good mapping from age to plasticity. For the experiments reported here,  $\delta_{age} = 4$ .

For the environment and network presented here, it was found that a range of numbers from -0.25 to 0.25 was a good selection for the available values for  $\bar{\eta}$ . This gave the evolutionary algorithm the opportunity to tune the learning rate quite finely, as well as having the possibility to move fairly rapidly towards high or low rates when needed.

Negative learning rates are not allowed. If the learning rate goes into negative values, it is treated as a learning rate of zero. Still, the negative value of the learning rate is remembered when calculating the learning rate for the next age. This makes it possible for evolution to drive the learning rate far into the negative domain, meaning it can effectively “shut off” learning by adjusting the learning rate. This way, the encoding allows evolution to generate sensitive periods that are quite resistant to disruptive mutations.

## Evolutionary Parameters

Table 1 gives the parameters of the evolutionary algorithm for the experiments reported here. Crossover probability gives the probability of crossover *per individual*, and mutation probability gives the probability of mutation *per bit* in the individuals’ genotype.

The same parameters were used for all runs, except for the case where *static* learning rates were evolved. For these runs, we needed only *three* genes to specify the learning strategy followed by an individual (one gene per learning link). Those three genes were each encoded by 20 bits, to allow evolution to fine-tune the static learning rate. The remaining parameters were not altered.

## Results

### Learning costs and sensitive periods

Figure 4 shows average evolved age-dependent plasticities for the three different behaviors in the task when there is no cost of learning. As expected, no sensitive periods are

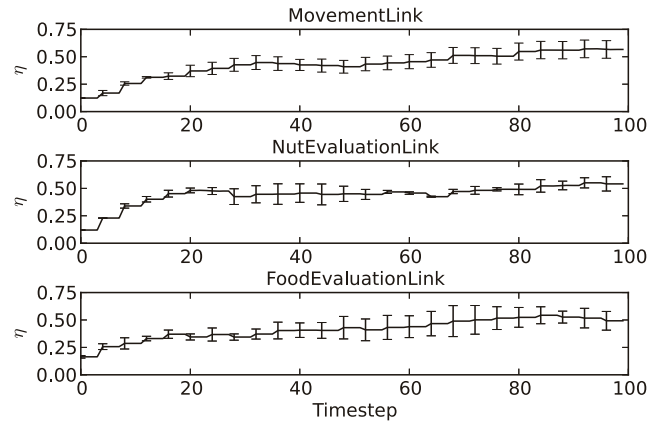


Figure 4: Evolved plasticities with no cost of learning. The figure displays the plasticity in the three links of the network as a function of the agent’s age. – Averages over 50 runs. Error bars show one standard deviation.

formed, as there is no pressure to form them imposed by the evolutionary algorithm.

A cost of learning on the fitness of individuals is documented in biology (see for instance Mery and Kawecki (2003) for an example from the common fruit fly). We simulate the costs associated with plasticity to see how this affects the evolution of age-dependent plasticity. The cost of learning is implemented as a term subtracted from an individual’s total fitness. It is proportional to the sum of areas under the agent’s three plasticity graphs.

Figure 5 shows the evolved plasticities when adding this cost to fitness evaluations. As the figure shows, we see cascading sensitive periods moving from lower to higher levels of behavior. Averaging over 50 individuals smooths the sensitive periods, meaning the cascades show more overlap than what is normally present in individuals. We will see later in this section that the evolved plasticities within individuals do show the expected ordering of learning.

Another reason for the large degree of overlap is that the individual behaviors are very simple to learn: they can be learned by a single observation of the relevant association. Had each behavior been more complex, taking more time to learn, it is expected that sensitive periods would be further separated in time.

**Sensory deprivation** A common way of studying sensitive and critical periods, is to subject individuals to sensory deprivation up until a certain age, and study their development after this age. If they are unable to follow a regular path of development, we have an indication that the plasticity of the considered system underwent a critical period before that age. In our deprivation study, we subjected the winner individual of each evolutionary run to sensory deprivation on all three levels of behavior, and measured its performance on the considered behavior. For instance, to test

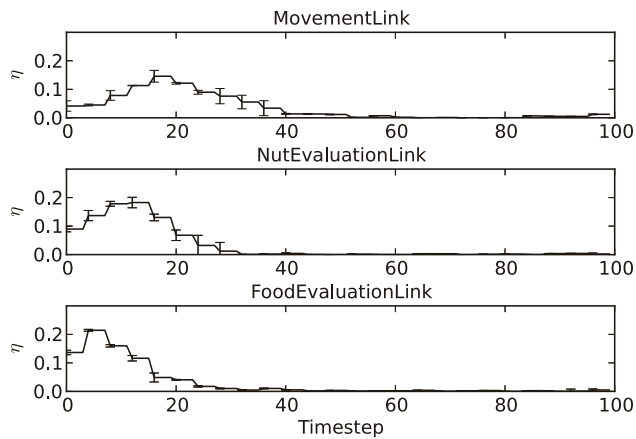


Figure 5: Evolved plasticities with a cost of learning – Averages over 50 runs. Error bars show one standard deviation

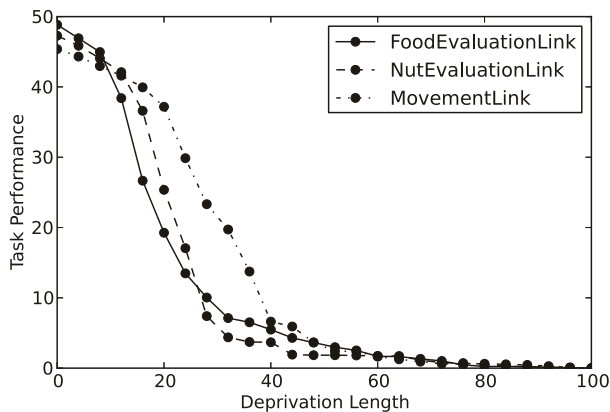


Figure 6: The result of sensory deprivation. – Averages over 50 runs. Dots show measured values, lines interpolate between them.

whether there was a sensitive period for learning about nuts, we waited for a given time before presenting the individual to nut stimuli, and measured how well it was able to learn the nut association task from this age.

Task performance was measured as the amount of correct associations made during the rest of the individuals' life (after finishing sensory deprivation), reduced by the amount of incorrect associations made. An inverse relationship between deprivation length and task performance was observed for individuals with a constant learning rate throughout life, because for longer deprivation lengths, individuals had less time to accumulate performance points.

For individuals allowed to evolve age-dependent plasticity, the sensory deprivation resulted in the task performance shown in Figure 6. All behaviors approach zero for sensory deprivation above a certain age, indicating that learning the associated behavior was not possible after this age. Further, we see the cascading of learning observed in Figure 5 af-

fect the timing of the age when sensory deprivation disrupts further learning.

The fact that performance seems to fall off *gradually* with increasing durations of sensory deprivation is due to differences in the evolved timing of sensitive periods between individuals. Studying single individuals, we observed that the cut off is much more dramatic. In other words, the term *critical period* describes this learning better than *sensitive period*. The reason is of course that *age* is the factor controlling plasticity, so delaying stimuli beyond a certain age will prevent all learning.

## Learning Order

This section presents statistics about the *ordering* of sensitive periods. We evaluated 50 individuals, noting how many of these formed the *expected order* of sensitive periods. The order of sensitive periods was measured by calculating the *center of mass* for plasticity in each behavior, and ordering them in order of ascending center of mass. An early center of mass for a behavior means a lot of learning is going on early in the life of the agent, corresponding to an early sensitive period.

The order of the centers of mass was then compared to the ordering that would be generated by random processes. For three different behaviors, 6 different orderings are possible, and the expected number of times each would be seen is  $1/6 * 50 = 8.33$ . For a statistically significant indication that the expected ordering is preferred, we need to see the expected order of sensitive periods at least 14 times, which is associated with a p-value lower than 0.04. For comparisons between *pairs* of behaviors, we need to see the expected order of sensitive periods at least 32 times, again indicating a p-value lower than 0.04.

Table 2 shows statistics from runs of the experiment with and without a learning cost. The table shows that there is *no significant evidence* that plasticity schedules learning in the expected order, when there is *no cost of plasticity*. However, when individuals are evolved *with* a plasticity cost, they are forced to being more cost-effective, thus scheduling their learning.

**Shuffling** After observing that sensitive periods tended to form in an order from “lower” to “higher” levels of behavior, we wanted to test just how important this ordering was. To do this, we shuffled sensitive periods in each of the winner individuals from our 50 runs of evolution. This was done by extracting evolved learning strategies from one layer, and inserting it into another within the same individual. The resulting individuals were each tested on 50 new mazes, and their average fitness was stored as an indication of how well they were able to learn preferences in these mazes. Notice that shuffling does not affect the learning costs of an individual, so a difference in fitness indicates a difference in learning performance.



Learning Cost	p(F-N-M)	p(F-N)	p(N-M)	p(F-M)
Yes	< <b>0.001</b>	<b>0.008</b>	< <b>0.001</b>	< <b>0.001</b>
No	0.317	0.839	0.556	0.336

Table 2: The ordering of sensitive periods. Letters F, N and M are used to indicate the different learned associations: **F**ood, **N**uts and **M**ovement associations. The following columns show the probability of observing the evolved ordering of sensitive periods by chance, first for all three behaviors, then their pairwise orderings. Results in bold signify statistically significant evidence that evolution prefers to arrange sensitive periods in the expected order. All results were obtained by doing 50 runs of the EA.

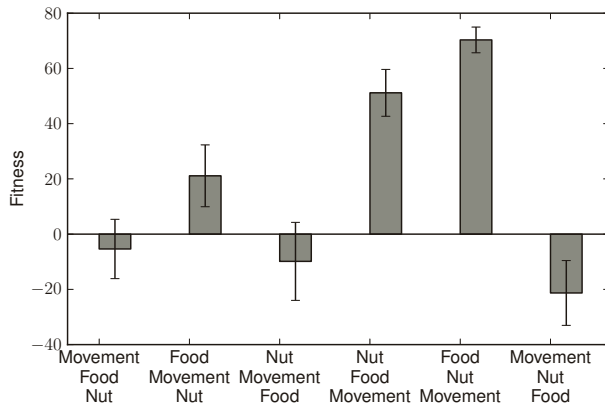


Figure 7: How learning is affected by shuffling sensitive periods. – Averages over 50 runs. Error bars show a 95% confidence interval of the mean.

Figure 7 shows the result of the shuffling of sensitive periods. The labels below axes indicate what new ordering of learning the shuffling corresponds to. (Food - Nut - Movement) was the original ordering, so this label indicates no shuffling. We see that increased shuffling gives a decrease in learning ability (seen as a decrease in fitness). Shuffling only two of the evolved sensitive periods, gives less of a fitness decrease than shuffling all three. The lowest fitness is observed when the sensitive periods are completely reversed, indicating that a proper sequencing is essential for learning in the winner individuals.

### The utility of age-dependent plasticity

So far, we have seen how age-dependent plasticity allows the formation of sensitive periods. In this section, we will analyze exactly what is the *utility* of these periods. We will do that by comparing the fitness of evolved agents with static plasticity levels and with plasticity levels that vary throughout life.

Figure 8 shows the fitness of individuals *with* and *without* a plasticity cost. During evolution, the plasticity cost was active in both cases, meaning all individuals evolved to form as cost-effective learning strategies as possible. As the right figure shows, the agents allowed to utilize age-dependent

plasticity become more *cost effective* – their fitness is higher because they pay a lower cost. However, as can be seen in the left figure, this more efficient learning strategy does not degrade their learning ability. When not applying a cost of plasticity, the fitness is only based on what the agents can learn, and this value is not significantly different for the two types of learners.

### Conclusion

In this paper we have shown how sensitive periods in learning can emerge for a reinforcement learning task by allowing an evolutionary algorithm to tune the mapping from ages of individuals to plasticity values. The task to be solved required several levels of interdependent behavior to be learned, and a different age-plasticity mapping was evolved for each level. The evolved sensitive periods showed the ability to *sequence* learning in a bottom-up fashion, allowing the network to learn the simpler behaviors first, before learning the higher-level behaviors that depend upon them.

A condition for observing the sensitive periods was that plasticity had an associated *cost*. This cost made evolution form solutions that learn in a cost-effective way. It is the balance between having the ability to learn while paying as low a cost as possible, that drove evolution to finding *cascading sensitive periods*.

In summary, we have seen how evolved sensitive periods can *sequence* the learning of sub-behaviors, and that the factor that drives this sequencing is achieving a good balance between the costs and benefits of learning. Currently, we are working on extending this model to account for experience as a regulator of plasticity, and to include the possibility of evolved hard-coded preferences.

Another interesting direction for future work is to investigate critical periods in more complicated tasks, perhaps tasks that cannot be solved with a constant learning rate. In other words, tasks that would normally be approached with a different technique such as incremental learning. We believe sensitive periods could be useful in such tasks, because they offer a way of scheduling learning activities between different behaviors.

Finally, the complexity of individual behaviors in the current study is quite low. It would be interesting to see how sensitive periods are affected by scaling up the complexity

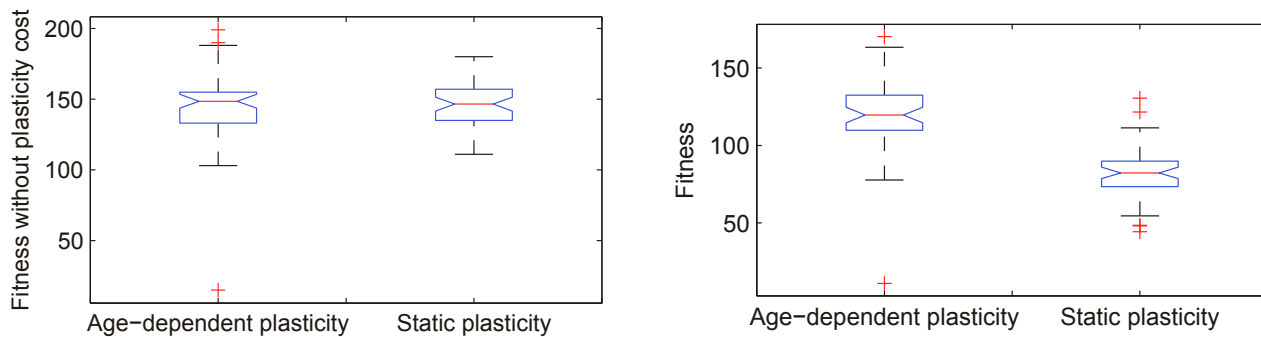


Figure 8: Box plots of fitness values calculated with and without plasticity cost. Nonoverlapping box notches indicate significantly different means with a 0.05 significance level. – Averages over 50 runs.

of behaviors, and also how they are affected by tasks where the progress of learning each individual behavior is less predictable.

## References

- Armstrong, V. L., Brunet, P. M., He, C., Nishimura, M., Poole, H. L., and Spector, F. J. (2006). What Is so Critical?: A Commentary on the Reexamination of Critical Periods. *Developmental Psychobiology*, 48(May):337–44.
- Bullinaria, J. (2009). Lifetime learning as a factor in life history evolution. *Artificial Life*, 15(4):389–409.
- Bullinaria, J. A. (2003). From biological models to the evolution of robot control systems. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 361(1811):2145–64.
- Burghardt, G. M. and Hess, E. H. (1966). Food imprinting in the snapping turtle, *Chelydra serpentina*. *Science*, 151(7):108–109.
- Cashdan, E. (1994). A sensitive period for learning about food. *Human Nature*, 5(3):279–291.
- Darmaillacq, A.-S., Chichery, R., and Dickel, L. (2006). Food imprinting, new evidence from the cuttlefish *Sepia officinalis*. *Biology letters*, 2(3):345–7.
- Downing, K. L. (2010). A Script-Based Approach to Evolving Neural Networks. In *Proceedings of the second Norwegian Artificial Intelligence Symposium*, pages 29–36.
- Fellous, J.-M. and Linster, C. (1998). Computational models of neuromodulation. *Neural computation*, 10(4):771–805.
- Harwerth, R. S., Smith, E. L., Duncan, G. C., Crawford, M. L. J., and von Noorden, G. K. (1986). Multiple Sensitive Periods in the Development of the Primate Visual System. *Science*, 232(4747):235–238.
- Hensch, T. K. (2005). Critical period plasticity in local cortical circuits. *Nature Reviews Neuroscience*, 6(11):877–888.
- Hubel, D. H. and Wiesel, T. N. (1970). The period of susceptibility to the physiological effects of unilateral eye closure in kittens. *Journal of Physiology*, 206(2):419–436.
- Hurford, J. R. (1991). The evolution of the critical period for language acquisition. *Cognition*, 40(3):159–201.
- Johnson, M. H. (2005). Sensitive periods in functional brain development: problems and prospects. *Developmental psychobiology*, 46(3):287–92.
- Kirby, S. and Hurford, J. R. (1997). The evolution of incremental learning: language, development and critical periods. Technical report, Language Evolution and Computation Research Unit, University of Edinburgh.
- Knudsen, E. I. (2004). Sensitive Periods in the Development of the Brain and Behavior. *Journal of Cognitive Neuroscience*, 16(8):1412–25.
- Lewis, T. L. and Maurer, D. (2005). Multiple sensitive periods in human visual development: evidence from visually deprived children. *Developmental psychobiology*, 46(3):163–183.
- Mery, F. and Kawecki, T. J. (2003). A fitness cost of learning ability in *Drosophila melanogaster*. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1532):2465 – 2469.
- Punzo, F. (2002). Food imprinting and subsequent prey preference in the lynx spider, *Oxyopes salticus* (Araneae: Oxyopidae). *Behavioural Processes*, 58(3):177–181.
- Soltoggio, A., Bullinaria, J. A., Mattiussi, C., Dürr, P., and Floreano, D. (2008). Evolutionary Advantages of Neuromodulated Plasticity in Dynamic, Reward-based Scenarios. In *Artificial Life XI*, volume 11, pages 569–576.
- Trotter, Y., Gary-Bobo, E., and Buisseret, P. (1981). Recovery of orientation selectivity in kitten primary visual cortex is slowed down by bilateral section of ophthalmic trigeminal afferents. *Developmental Brain Research*, 1(3):454–450.
- Werker, J. F. and Tees, R. C. (2006). Speech perception as a window for understanding plasticity and commitment in language systems of the brain. *Developmental Psychobiology*, 46(3):233–51.

# Autopoiesis Facilitates Self-Reproduction

Jean Sirmai

Clinician and Hospital Medical Practitioner, Ile de France.

[jean.sirmai@orange.fr](mailto:jean.sirmai@orange.fr)

## Abstract

The first *in silico* models of self-reproduction only focused on the logic of the mechanisms that execute and copy the genome or renew the membrane, but neglected associated physical constraints. This may have resulted from modeling through cellular automata, which are unable to represent the cohesion of objects in movement and interaction. In previous work I presented a new, well structured and powerful tool based on a graph rewriting system embedded in a spatial automaton. This tool employs combinations of a unique symbol and can represent an unlimited variety of moving and interacting objects. As transitions are local and occur at random, each trajectory of the system differs. However, dependent events can always be represented in their natural order.

With this tool, I built a representation of an autopoietic individual. More recently, I hypothesized that this model could also be used to demonstrate self-reproduction because most of the mechanisms required for growth are already available in the autopoietic individual and few additional functions are needed. Here, I report the advancement of the model to demonstrate the ability of the autopoietic individual to self-reproduce. During self-reproduction, autopoiesis remains active and the lifespans of the various components are unchanged.

Pathological morphologies can be observed when some metabolic pathways are disturbed. Using appropriate approximations, some thermodynamic parameters can be evaluated. Additionally, a second autopoietic and self-reproductive individual can be represented within the same environment. Further, the model could be used to describe the space phase domain and invariants characteristics of each of these individuals, whose systematic enumeration and classification can be envisioned. Based on this model, I propose that autopoiesis facilitates self-reproduction.

## Introduction

An entity capable of self-renewal is said to be autopoietic (Maturana and Varela, 1973). Commonly, biological entities (e.g., cells, tissues, societies) are observed to be able to generate almost all the components with which they maintain their structure and functions. I hypothesize that this can result from the association of two independent properties: persistence and cohesion.

Persistence is the property of the entities that are able to maintain their composition while constantly renewing themselves. Under this property, each part of the entity is produced by at least one transformation and destroyed by another; these transformations regulate one another<sup>1</sup>. Such

entities depend on a permanent input and output of energy and materials. The ingoing components are rich in potential energy, while the outgoing components are poor. External components can be classified as resources, neutrals, or toxins. When exposed to a toxin, the whole may be able to compensate for its effects. If it is not and this results in the defect of a major regulatory pathway, it may not remain persistent. It controls its composition, which fluctuates around a mean, but not its shape and size, which depend on the limits provided by its environment. Its lack of cohesion hampers its transfer in another environment. It can split in two persistent entities if each resulting part keeps the initial composition and is provided with input and output pathways, but it cannot control this process and self-reproduce. Conversely two persistent entities of compatible composition in direct contact with each other can merge. Persistence could have been a property of some instances of the “prebiotic soups” imagined by Oparin and Haldane (see Popa, 2004).

A persistent entity is autopoietic if some of its components, other than the entering and outgoing ones, ensure its cohesion. Compared to a simply persistent entity, an autopoietic individual is endowed with several new properties. First, it controls not only its composition but also, at least partly, its limits, inputs, outputs, shape, and size. It can keep its shape longer than the parts composing it can. Second, in so much as its state remains stable, its entropy remains roughly constant while that of its environment increases (Schrödinger, 1944). Third, the more energy is available in its environment the more it controls its use of this energy (Virgo, 2011). Fourth, it can attain a maximal performance in extraction and use of energy from its environment. Conversely, pathological states exist where its global performance is reduced. Fifth, it can be moved and then maintain itself in any non-toxic environment providing only its inputs. Sixth, it can be associated to self-reproduction. Seventh, it loses these properties if split (giving rise to the etymology of the word “individual”). Correlatively, two similar autopoietic individuals in direct contact with each other will not merge (McMullin, 2004). Cohesion can be obtained by including all the components in a compartment. Another possibility would be to link them all together. The existence of such a kind of individual would demonstrate that the presence of an interior milieu is not a necessary condition for autopoiesis.

Prior work has proposed that self-reproduction is a particular case of self-production (Sharov, 1999). In self-reproduction, an individual is able to extract some energy and matter from its environment and use it to produce a new individual that is similar to itself and that remains distinct. Once the reproductive process has been completed the new individual cannot be distinguished from the other by anything but their history. The model I describe here supports the hypothesis that autopoiesis can facilitate self-reproduction and show the details of this process in the case

<sup>1</sup> It identifies each part of itself at least twice: once to synthesise it and once to destroy it. We might assume two relationships exist, such that, 1) the more complex a component, the more efficient it is, and 2) the more complex a component, the more complex the operations required to build or destroy it. Then, because they are constrained to operate constantly on one another, all components of one persistent object will tend to share similar levels of efficiency and complexity.

of self-reproduction by budding. As most of the mechanisms required for growth are already available in an autopoietic individual, few additional functions are needed for this individual to reproduce itself.

Autopoiesis and self-reproduction were first described in bottom-up models (see Discussion). Recently, a first top-down model based on ordinary differential equations was described (Karr et al., 2012). This model is a proof of concept. It shows that some properties of a real living object can be computed. However, not all these properties can be simultaneously represented in detail. This is due to the impossibility of completely isolating an object as well as to a lack of knowledge and of computing power (Zwirn, 2000). Karr's model represents both some biochemical mechanisms (non-biological *stricto sensu*) and some biological properties. These are mainly the reproduction cycle, autopoiesis, and the energetic balance of *Mycoplasma genitalium*. As it uses successive approximations of several sub-models, this model is redundant. The redundancy contrasts with the minimal expression of the same properties in the bottom-up models. However, one can expect that top-down and bottom-up approaches (analytical and synthetical) will converge towards one another (Hucka et al., 2003). Bottom-up models could help to define properties and to extract only the meaningful information relevant for each of them from top-down models.

Anatomical characteristics of life have been beautifully described by Goodsell (see images following References) (2009). His work seeks an integrated view of all the components of a cell. He erases the mechanistic details that would be required for a comprehensive description of all the functions of those components and extracts only those that enable the approximation of the main anatomical and physiological properties of the living. Thus, his drawings inspire this work.

## Methods

The previously described platform associates a graph rewriting system to a spatial automaton and provides a new, well structured and powerful language to represent almost any biological phenomena (Sirmai, 2011)<sup>2</sup>. It can be seen as a new artificial chemistry (Dittrich, 2009) that can apply to any phenomenon characterized by a great variety of forms and interactions. This diversity suggests the use of a combinatorial method. To enable such a possibility, I introduced indexes (previously called "links") in the cells of a spatial automaton. An index belongs to a cell and points to a neighboring one. A set of cells pointing to one another by their indexes is an object. The cells of the state containing indexes then become nodes of a graph and the indexes are the edges. These edges are oriented and weighted, as many indexes can point towards the same neighbor.

Each object is an isolated part of the graph. It is completely described by the location and orientation of the indexes that compose it. This formalism does not limit the size, shape or number of objects. In the present model, no index may designate an empty cell and only adjacent cells can be designated. These parameters could be modified.

Each transition associates a set of conditions to a set of operations. All the conditions are the same type: they test

the number of indexes in a given location and orientation. All operations are of the same type: they move an index from a place to another. This formalism does not limit the variety of movements or transformations that can apply to each object or couple of objects. In the present model, no operation changes the total number of indexes arranged in space or the number of indexes of a cell.

The automaton deals only with indexes in the cells. It uses no conditions on the objects such as a name or a color. Users can recognise the objects drawn using these indexes (Figure 1, left panel) but this recognition is made easier by the use of colors (Figure 1, right panel), as can be seen by comparing the two panels (Fig. 1), which display the same workspace. A second-level language may be superimposed to the first to recognize the objects and enable the user to interact directly with them rather than with the indexes. In the present model, the workspace is a two-dimensional hexagonal matrix without boundaries, wrapped over a torus.

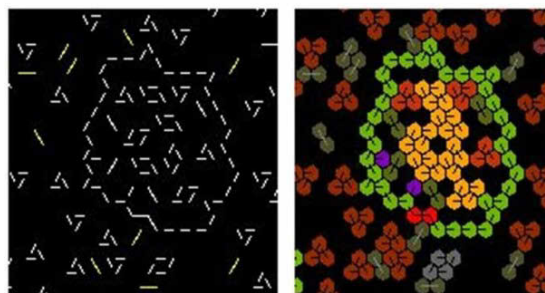


Figure 1: Two views of the same space state.  
Left, indexes alone. Right, colored particles.

The space is not explored using its coordinates but according to its content. Each transition converts only a part of the space. This part is centered by a randomly chosen index. It is then assessed through different sets of conditions. If a set of conditions is satisfied, then the set of associated operations is performed.

All information regarding the description of objects is in the space. All information concerning their movements and interactions is in the transitions. No other information is encoded.

Transitions can move, deform, transport, or transform objects. Displacements, deformations, and transports maintain the objects in the same class. Transformations shift them from one class to another. Each class is associated with some characteristic pattern of indexes that can be identified by some adequate conditions' sets.

Here, interactions are not associated with one object, but with at least two and possibly more. They are thus described only once. The downside is that each transition must identify each object involved (pattern recognition). Each transition can apply to the few neighboring cells representing one object to move or to parts of two objects in interaction and not necessarily to the entire space at one time.

The objects to which transitions apply are chosen at random. Transitions occurring in a random order adequately represent independent events. Yet, it happens that an event depends on another one which determines it, and the determining event always occurs always before the dependent one. In the same way, the transition representing

<sup>2</sup> The open-source program is available at [www.interactor.fr](http://www.interactor.fr).



the determined event must always occur before the transition representing the dependent event. This can be achieved by using an intermediate state, which is the result of the first transition and the beginning of the second. There must be no transition going directly from the initial state of the first transition to the final state of the second. If this rule is followed, the second transition will always occur after the first, although the order in which transitions occur is chosen at random.

Hereafter, the environment will be hidden to concentrate on the description of the individuals. Importantly, some space remains always free in the compartment and outside enabling a permanent random movement of all objects.

## Results

Two individuals will be described: one called “Tiuccia” and the other “Lagny”<sup>3</sup>. When the model is running, Tiuccia is easily recognizable as it is circled in green and Lagny looks like a yellow little worm.

Tiuccia is made of seven varieties of aggregates: five are implied in autopoiesis, two in reproduction. Tiuccia comprises a membrane enclosing an internal compartment. The membrane ensures the cohesion of the whole. It is made of one-index particles pointing each to the next one. Because of its asymmetry (all its indexes are oriented clockwise) its inside and outside faces can always be locally recognized. The internal compartment contains freely-moving tetramers, trimers, and dimers (Figure 2). Trimers and dimers are also present in the environment (not shown in this view). An arbitrarily high potential energy is assigned to the trimers (food) and a low one to the dimers (wastes). The membrane is selectively permeable: trimers can only enter (**te**; See Table 1) and dimers can only exit (**de**). In the environment a mechanism converts permanently dimers in trimers to maintain a favorable condition.

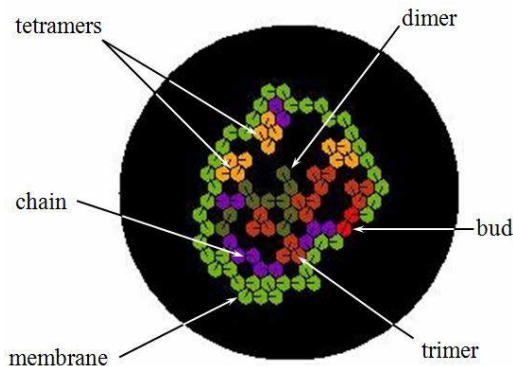


Figure 2: Anatomy. Screen capture of Tiuccia indicates the objects it comprises.

Small chains of one, two, three, or more units long are attached to the internal face of the membrane. When the membrane wrinkles near an attached chain, one of its units can be removed and transformed into a chain unit. This transformation (**mc**) lengthens the chain while the membrane shortens. The membrane's continuity remains

ensured. When a chain is at least four units long, its terminal end can fold in on itself and transform into a tetramer (**ct**). Therefore, the presence of tetramers indicates that some membrane catabolism (destruction) occurred. When the tetramer concentration increases, tetramers catalyse their own catabolism. If three tetramers are adjacent, the central one will be transformed into two dimers (**td**).

When two trimers are close to the membrane, one of them catalyses the transformation (**tc**) of the other into a one-unit chain and a dimer. The one-unit chain is attached to the internal face of the membrane and will lengthen as previously described. The presence of many trimers in the cell is an indicator of a high level of accessible food. This signal initiates membrane catabolism.

When a tetramer and a trimer are close to the membrane, the tetramer catalyses the transformation (**tm**) of the trimer in a unit of membrane and a dimer. The unit is inserted in the membrane. Tetramers are a signal of a previous membrane catabolism and a condition of its anabolism (synthesis).

Because of this coupling of regulation, the size and shape of Tiuccia remain almost constant while all its components are permanently renewed. Inputs and outputs are in competition with the synthesis and destruction of the membrane that regulate its length. Figure 3 and Table 1 depict this metabolism.

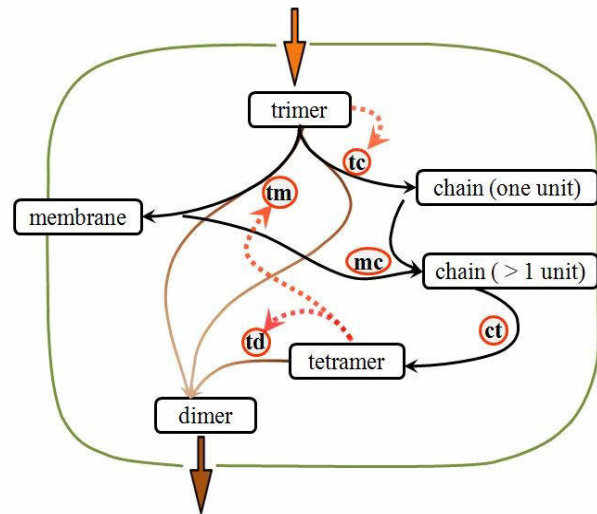


Figure 3: Metabolic pathways and their regulations.

## Budding

The “budding” process occurs through ten transformations and transports. Only four of these are specific to reproduction. The other six occur in both autopoiesis and reproduction, and their denominations remain unchanged in the following description.

The first step of budding consists of the capture by the bud of a double-index dimer moving near it in the environment (Figure 4a; budding transformation 1). The bud itself is made of particles containing two indexes. The captured dimer is integrated close to the bud into the membrane and the whole gives rise to a 4-double-index particle sequence: the cord.

<sup>3</sup> From the names of the cities where they were first observed.

Name	Description (effect)	Conditions of realisation and metabolic meaning
te	trimer entry	trimers are present in the environment; free space is available inside the membrane is flexible; no chains are attached locally
tc	one trimer $\rightarrow$ one dimer + new chain (one unit)	presence of another trimer means that a high food content is available the new chain is anchored to the internal face of the membrane
mc	membrane (one unit) $\rightarrow$ chain (one unit)	the membrane must wrinkle towards inside the chain must be attached to that part of the membrane
ct	chain (four units) $\rightarrow$ one tetramer	the chain length is almost four units long it can fold in on itself at random
tm	trimer $\rightarrow$ one dimer + membrane (one unit)	the new unit is inserted in the membrane; this transformation depends on the presence of a tetramer that acts like a catalyst and is left unchanged
td	one tetramer $\rightarrow$ two dimers	two other tetramers are adjacent and act like catalysts
de	dimer exit	free space is available outside; the membrane is flexible no chains are attached to the membrane locally

Table 1: Metabolic pathways. The names refer to the names in figure 3 and in the text.

The cord catalyses the second step (Figure 4b): it captures a tetramer and transforms it in a short membrane fragment (budding transformation 2). This fragment is located outside of the main compartment and attached to it by its two extremities. Tetramers can be found in the cell only and they are a sign of its maturity and good nutritional status. The position of the cord between the two compartments that it both links and separates is asymmetric at its insertions. The main compartment will be called the parent and the smallest the offspring.

The asymmetry enables the cord to specifically catalyse the elongation of the new fragment of membrane starting from trimers provided by the parent (budding

transformation 3). A particle of a parent trimer close to the cord insertion is added to the offspring's membrane. The remaining dimer stays in the parent and will be eliminated later (**de**).

These three transformations are enough to initiate a complete new autopoietic process that enables the growth of the offspring. The next steps are part of the autopoietic process. They occur when the offspring's membrane becomes long enough to absorb trimers (Figure 4c; **te**) and to release dimers (**de**). As soon as two trimers have been absorbed, they can give rise to a one-unit chain (**tc**). As the offspring's membrane keeps growing, due to the parent's assistance, the chain can lengthen (Figure 4d; **mc**).

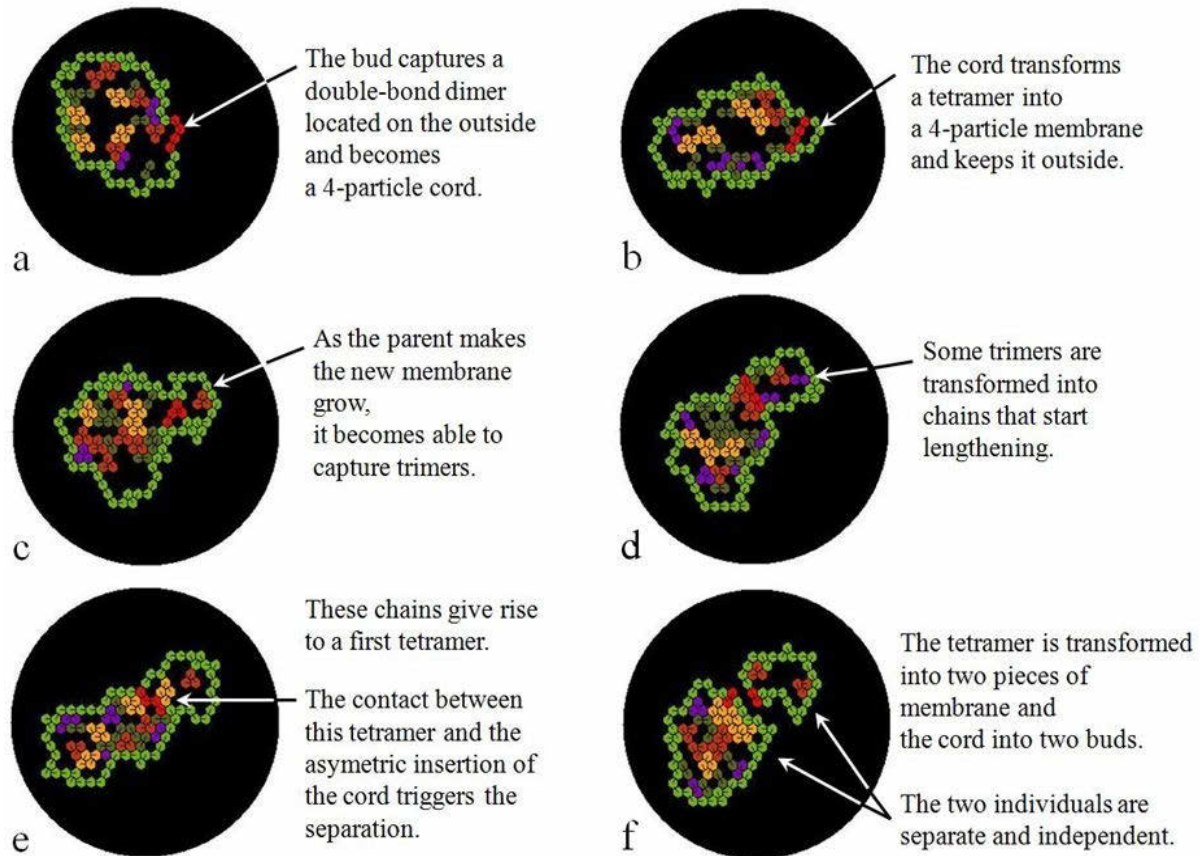


Figure 4: Budding.

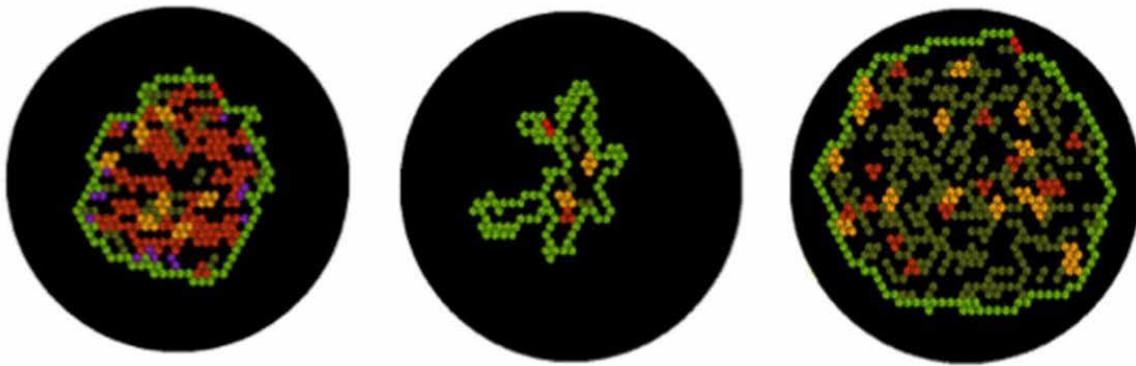


Figure 5: Three abnormal morphologies associated each with a different metabolic defect.

When it attains a four-unit length, its transformation in a tetramer becomes possible (Figure 4e; **ct**). The production of a tetramer by the offspring is a signal of maturity. It guarantees that the membrane has grown enough to ensure first its own degradation and second its own production. This signal is recognized by the cord and initiates the separation (Figure 4e; budding transformation 4). The cord divides in two buds of two particles each, one belonging to the parent and the other to the offspring (Figure 4f). These individuals become completely independent.

As an offspring's tetramer is transformed into membrane during the separation, the offspring may be found to have only one tetramer or none at this stage. This feature is characteristic of a young individual and will not persist. Other tetramers will be produced continuously, and, once there are two tetramers, the number will not decrease anymore because the presence of two tetramers is required to catalyse the destruction of a third one (**td**). The two individuals produced will then remain completely similar and only distinguishable from one another by their history.

As long as the environment remains atoxic and provides the required resources the autopoietic and self-reproduction processes never stop.

### Pathology

For some given values of the metabolic fluxes, the individual seems to remain in a basin of attraction. The measured lifespans of each of its components appear always in the same characteristic distribution. The histograms describing the distributions of their quantities are the same, and the correlations between are also unchanged. Of course, these observations are not a demonstration that this will always be the case.

Modifications of the flux of the metabolic pathways result in various morphological changes. Indeed, for each different metabolism a different shape results. Three examples of such morphologies are presented here (Figure 5) to demonstrate the diversity of the patterns that can be obtained. Reciprocally, modifications of the individual's shape can result in modifications of its metabolic fluxes.

The present set of regulations of Tiuccia enables always its total recovery. In some extreme cases, its metabolism can almost be stopped but no lysis or apoptosis phenomena can be observed. These phenomena constitute new properties and require the addition of new metabolic pathways to be attained.

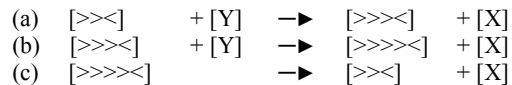
### Description of "Lagny"

Lagny is a short chain of one-index particles each pointing towards the next. Only the last one points back towards the previous.

The whole looks like this:  $[>>>><]$ .

Let's call  $[><]$  the "head" and  $[>>>]$  the "tail".

The length of Lagny varies cyclically. Only three transitions are possible. During transitions (a) and (b), the head  $[><]$  "eats" a trimer  $[Y]$ . Trimers are rich in potential energy. "Eating" a trimer means releasing a dimer and keeping one unit of the trimer, which becomes a part of the head. Another part of the head is also transformed and lengthens the tail. The dimers  $[X]$  are poor in potential energy. Transition (c) releases a dimer from the end of the tail when it is long enough. The whole reaction set could be written as follows:



As each of its parts is continuously renewed by itself and it always keeps its cohesion, Lagny is an autopoietic individual.

Lagny can reproduce itself when it eats a dimer made of two two-link particles,  $[2-2]$ . This can happen only when its length is maximal. Once ingested, the  $[2-2]$  dimer transforms the beginning of the tail into a new head and settles between it, the one-unit tail, and the other head. The two heads continue eating as usual, and  $[2-2]$  serves as their link with each other and the tail. But, as the two new bodies keep growing,  $[2-2]$  is progressively pushed back towards the tail. Finally, when the two new bodies are long enough, the ancient tail completely disappears and a  $[2-2]$  dimer is expelled. This separates the two new similar Lagnys, which then continue independently.

As two identical shapes would have the same potential energy, nutrients, wastes and all their metabolic intermediates must have different shapes. These shapes are pure conventions and can be changed. Other components or transformations can be added. The following rules, however, must always be applied:

- If two components have different potential energy they must have different shapes.
- Each component of the individual must always be destroyed and renewed.
- Cohesion of the individual must always be ensured.



## Discussion

This model demonstrates that autopoiesis and self-reproduction are not only compatible, but that in the case of budding, the first can facilitate the second. It also confirms that bottom-up models, that were first used to analyse complex biological objects by representing and defining some of their functions, can now be used to synthesise new objects associating several of these properties. Additionally, in this model, pathological morphologies can be observed when some metabolic pathways are disturbed. Finally, several individuals of different natures can be simultaneously represented interacting within the same environment. The abstract characteristics of this model allow not only biochemical but also robotic, nano-physical, or other interpretations.

### Comparison to Other Models of Self-reproduction

In other models of self-reproduction, the individuals produced are not exactly similar and they are not autopoietic (Hutton, 2007). In some, they are not clearly separated (Ono and Ikegami, 2000). Some require the association of two different formalisms to represent movements and transformations (Smith et al., 2002) or small and large objects (Wishart et al., 2005). Others, partly because they use CA<sup>4</sup>, are neither autopoietic nor mobile (von Neumann, 1966; Langton, 1986; Sipper, 1998; Ishida, 2010). They invade space only during reproduction and then stay immobile. Some, like in molecular dynamic studies, are oriented towards the detailed description of a mechanism more than towards the integration of all of them to produce a biological property (Bersini, 2010). Swarm chemistries (Sayama, 2009) or diffusion–reaction models (Virgo et al., 2013) conform to some thermodynamic constraints but the “individuals” or their components do not have defined boundaries and anatomy. By contrast with agent based models, this formalism doesn't associate several functions to one object but several objects or parts of objects to one function. Functions are therefore described only once.

### Autopoiesis and Self-reproduction: Which Came First?

This model supports the hypothesis that, in the case of a budding mechanism (as opposed to division), a single structure could be sufficient to perform all of the mechanisms required for adding self-reproduction to autopoiesis. This structure would be synthesised when the individual's state indicates that it is healthy enough to reproduce itself. It would be able to separate and maintain linked a part of the membrane from the initial individual. When the inflow in the parent individual is sufficient, it would favour the use of a part of it to increase the size of this new membrane. As soon as this membrane has grown enough to perform its own entries itself, the autopoietic

regulations take over. Since the membrane is the energy entry point, the processes (metabolism and its regulation) that depend on the energy source will be activated. All the autopoietic regulations successively add up until the offspring has acquired most characteristics of the parent. The acquisition of the last of these characteristics would signal the separation. The structure that initiated budding disappears. Parent and offspring are identical. New budding processes can begin.

Can this relationship between autopoiesis and self-reproduction help us to understand how they appeared? According to the Oparin-Haldane hypothesis (see Popa, 2004), let's consider the case of a simply persistent individual capable of giving rise to an autopoietic individual. The transition to autopoiesis may, for example, be the creation of an isolate of the same composition as the parent but whose components are organized differently. The parts of this isolate are now linked together while it is still able to renew itself. We do not know if such a transition is a rare or common event since it does not produce a self-reproducing individual and, therefore, does not leave any trace.

Knowing that a single component may be sufficient to initiate and complete the process of self-reproduction, we can imagine that such a component was included in an isolate during a transition towards autopoiesis. If such an event occurred, it immediately created a new individual endowed with three major biological properties: autopoiesis, self-reproduction, and the ability to evolve.

If these properties did not arise simultaneously, one of them was acquired during the evolution of the other. However, only autopoietic individuals can acquire and control their self-reproduction. Persistent entities can only split and merge. Therefore, this model supports the hypothesis that self-reproduction was acquired either simultaneously with or after autopoiesis.

### Autopoiesis and Tolerance

The ability of autopoietic individuals to evolve depends on the control they exert on themselves. On one hand, this control enables them to better resist environmental variations than objects that are only persistent. On the other hand, if total, it may be an obstacle to any subsequent change. To acquire the ability to differentiate themselves from their parents, autopoietic individuals must be able to interact with new components. This implies an ability to tolerate some unexpected entries that are not constitutive parts of the individual.

Such a tolerance can seldom be passive, but most, if not all, known biological individuals are endowed with several active tolerance mechanisms, for example, redundancy of metabolic pathways, use of degenerate coding, compartmentalization, and ability to actively eliminate non-self components. This is associated with permanent identification of their self through an unceasing destruction and reconstruction.

The diversification of the entries has several consequences: it enables them to extract food from a greater variety of sources, and it allows new interactions with unknown foreign components which—even if insufficient—is a condition to the acquisition of new metabolic pathways including those required for self-reproduction.

Tolerance should be a target for future models.

<sup>4</sup> Cellular automata, because their rules modify only one cell's state, are unable to represent the coherence of objects in movement and interaction. Furthermore, starting from a given initial state, they calculate a determined unique trajectory and cannot show other possible evolutions and make probabilistic predictions. They are therefore more suitable for the description of an history than for prediction of possible future events. They can hardly be used to know what range of initial states and what kind of laws would have produced a given final state.



## Compatibility with Physical Laws

The model presented here allows an association of an entropy to each state space or part of the space. This entropy can be calculated by systematically enumerating all the possible distributions of the objects this space contains (complexions). Because each individual remains in a quasi-stable state in a basin of attraction, it is postulated that its entropy remains constant. Because the content of the environment is modified (for example, two trimers are removed while three dimers are added) it is postulated that the entropy of this environment increases.

To represent the conservation of energy, we must be able to calculate a quantity that remains constant from one transition to another. This quantity is equal to the sum of kinetic and potential energies of all the objects. A potential energy, a speed and a mass can be arbitrarily assigned to each object. From these data it is possible to calculate how the objects' speeds changed during conversions between potential energy and kinetic energy. However, in an asynchronous model, the speeds of all objects cannot be changed simultaneously and approximations are required<sup>5</sup>. It must be noticed that, due to the use of intermediate states (see methods), these approximations do not modify the order in which the dependent reactions occur.

Until a better approximation of kinetic energy is achieved, the only available reliable rule is that two identical shapes have the same potential energy. Therefore, nutrients, wastes, and, more generally, all their metabolic intermediates must have different shapes.

## Domain of the Individual in its Configurations Space

Here, the analogy between the biological reality and the model is consistent enough to justify the word "pathology". In both cases, we can distinguish a normal situation from several pathological ones. In "healthy" situations, the individual can attain its maximal performance in extraction and use of energy from its environment. In pathological situations, something is missing or in excess and the global performance is reduced. Each situation associates some anatomic (morphological) and physiological (functioning) characteristics.

In the case of Tiuccia, it has been observed that its mean renewal was faster with particular adjustments in the efficiency of some metabolic pathways. As this study is not extensive, it remains possible that better performance can be achieved with other adjustments.

Each possible morphology is a spot in the configurations space of the individual. This configurations space is an abstract space whose number of dimensions is equal to the number of classes of components of the individual. The individual is always in one place of this space. It can be in some places but not others. The set of places where it can exist constitutes its domain.

This model enables a systematic study of an individual's domain. This domain can be described spot by spot. However, by analogy with what is known of other physical (but not yet biological) systems, there may be approximations that allow more simple descriptions than

such an extensive enumeration. Such descriptions could rely on some characteristic parameters of the metabolism or associations of them. Some invariants might also be characteristic of the domain of each individual.

Another representation equivalent to the configurations space could be a transformations space whose number of dimensions would be equal to the number of transformations, each transformations varying in intensity.

## Towards a Systematic Enumeration of All the Life-as-It-Could-Be Forms?

It can be hypothesised that the domain description provides the most complete and simple representation of an individual as it comprises all its various metabolic states and the associated morphologies.

Domains descriptions could provide a method to distinguish an individual from one another. Given a set of constituents in a defined state and a set of relations between them, it should also be possible to know if they can constitute an individual or not.

These descriptions could also show how each individual is linked to its "relatives" and, therefore, open the way to a method allowing a systematic enumeration and classification of all the life-as-it-could-be forms (see Langton, 1986).

Finally, these questions are linked with epistemology since it can be hypothesized that, altogether, the description of all those life-as-it-could-be forms could constitute a Logical Tree of Life, independent of the Historical Tree of Life. All the statistical knowledge concerning reproducible biological phenomena should go in the first one; all the deterministic non-reproducible in the second.

## Conclusion

This model shows that, at least in the case of budding, autopoiesis facilitates self-reproduction. It analyses autopoiesis as an association of persistence plus cohesion and offers a rational definition of individuality. It supports the hypothesis that autopoiesis is simultaneous or precedes self-reproduction. It may be a guide towards a method allowing a systematic enumeration of all the life-as-it-could-be forms. It investigates the difficulties in representation of physical constraints and proposes some empirical rules.

The new platform used for these representations is a graph rewriting system embedded into a spatial automaton. It provides a simple and powerful language using combinations of a unique symbol to represent phenomena characterized by an unlimited variety of forms and interactions.

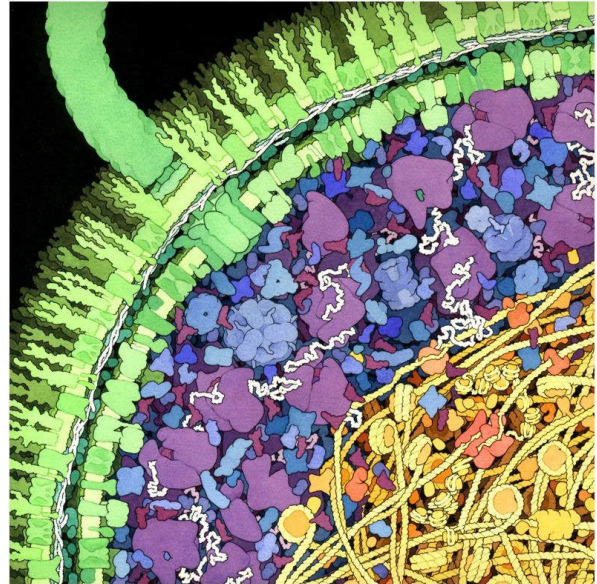
## Acknowledgements

I would like to thank Sébastien Boisard, Gerard Boyer, Christiane Mazurier, Sheila Cherry, Michel Morange, Barry McMulin, Hugues Bersini, Paul Bourguine, Chrystopher Nehaniv, Alexis Sharov, Tom Froese, Juan Carlos Letelier, Nathaniel Virgo, Peter Dittrich, René Doursat, Jacques Demongeot and Gerard Jagers op Akkerhuis for helpful discussion and insight.

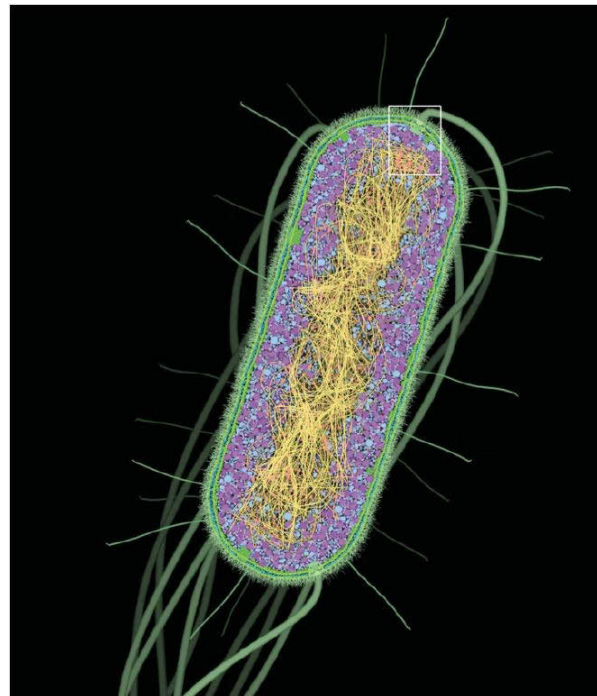
<sup>5</sup> A possible approximation that remains to be evaluated is to rank the objects according to each one's speed. Fast objects will then be treated more frequently, and the frequency of their interactions will be higher.

## References

- Bersini, H. (2010). Software replica of minimal living processes. *Orig Life Evol Biosph* **40**(2): 121-130.
- Dittrich, P. (2009). Artificial Chemistry. *Encyclopedia of Complexity and System Science*. B. M. e. al., Springer.
- Goodsell, D. S. (2009). Escherichia coli. *Biochem Mol Biol Educ* **37**(6): 325-332.
- Hucka, M., A. Finney, et al. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **19**(4): 524-531.
- Hutton, T. J. (2007). Evolvable self-reproducing cells in a two-dimensional artificial chemistry. *Artif Life* **13**(1): 11-30.
- Ishida, T. (2010). Simulation of cell-like self-replication phenomenon in a two-dimensional hybrid cellular automata model. *Journal of Robotics and Mechatronics* **22**(5).
- Karr, J. R., J. C. Sanghvi, et al. (2012). A whole-cell computational model predicts phenotype from genotype. *Cell* **150**(2): 389-401.
- Langton, C. G. (1986). Studying artificial life with cellular automata. *Physica D* **22**.
- Maturana, H. R. and F. Varela (1973). *Autopoiesis: the organization of living systems*. Dordrecht, Holland, D. Reidel Publishing Company.
- McMullin, B. (2004). Thirty years of computational autopoiesis: a review. *Artif Life* **10**(3): 277-295.
- Ono, N. and T. Ikegami (2000). Self-maintenance and self-reproduction in an abstract cell model. *J Theor Biol* **206**(2): 243-253.
- Popa, R. (2004). *Between necessity and probability: searching for the definition of life*. New York, Springer-Verlag.
- Sayama, H. (2009). Swarm chemistry. *Artif Life*(15): 105-114.
- Schrödinger, E. (1944). *What is life? The physical aspect of the living cell*. Cambridge [Eng.], The University press.
- Sharov, A. A. (1999). *Semiosis in self-producing systems*. Computing anticipatory systems CASYS'99- Third International Conference, Liege, Belgium, AIP Conference Proceedings.
- Sipper, M. (1998). Fifty years of research on self-replication: An overview. *Artif Life* **4**(3): 237-257.
- Sirmai, J. (2011). A Schematic Representation of Autopoiesis Using a New Kind of Discrete Spatial Automaton. *Advances in Artificial Life, ECAL 2011*, MIT Press.
- Smith, A., P. Turney, et al. (2002). JohnnyVon: self-replicating automata in continuous two-dimensional space. *NRC Publications Archive*.
- Virgo, N. (2011). Thermodynamics and the Structure of Living Systems, University of Sussex. **D. Phil.**
- Virgo, N., T. Froese, et al. (2013). The Positive Role for Parasites in the Origins of Life. *IEEE ALIFE 2013 Symposium*, Singapore.
- von Neumann, J. (1966). *Theory of self-reproducing automata*. Urbana, University of Illinois Press.
- Wishart, D. S., R. Yang, et al. (2005). Dynamic cellular automata: an alternative approach to cellular simulation. *In Silico Biol* **5**(2): 139-161.
- Zwirn, H. (2000). *Les limites de la connaissance*, Odile Jacob.



*Escherichia coli* (x 1 000 000 up and x 70 000 down) drawn by David S. Goodsell, Scripps Research Institute. Reprinted with permission from (Goodsell, 2009).



# Evolution of Social Representation in Neural Networks

Solvi Arnold, Reiji Suzuki and Takaya Arita

Graduate School for Information Science, Nagoya University, Japan  
solvi@alife.cs.is.nagoya-u.ac.jp

## Abstract

This paper describes an Artificial Life approach to Theory of Mind (ToM), the ability to employ mental representations of other minds in order to understand or anticipate the behaviour of others. We designed a model in which a population of neural network (NN) agents evolve the ability to predict, on basis of observation of past behaviour, others' future behaviour in novel circumstances. As agent behaviour is guided by private mental states, invisible to the predicting agent, this task forces agents to go beyond imitation and repetition of fit responses, requiring them to gain some degree of *insight* into the partner agent's internal configuration by observation of their externally visible behaviour. As such, this learning ability cannot be captured with conventional learning algorithms based on rewards or examples. We find that NNs equipped with neuromodulation mechanisms can be evolved to perform favourably on this task. The resulting networks are seen to behave as though they have a primitive form of first order ToM.

## Introduction

Theory of Mind (ToM) is the ability to employ mental models (representations) of other minds, in order to understand or anticipate the behaviour of others (Premack and Woodruff, 1978). The adaptive advantages of ToM are likely to be a driving factor in the evolution of cognition that recognizes others as well as itself as intentional agents. While ToM has become a hot topic in cognitive psychology and related fields, the phenomenon of "mirror neurons" (neurons that activate both when a given action is performed and when the same action is observed) has become a hot topic in cognitive neuroscience. Many researchers are intuitively inclined to link these two phenomena, thinking of mirror neurons as a neural basis for ToM, but the relation between the two remains murky. As such it seems potentially informative to try and evolve ToM-like abilities in neural systems.

Representation is a tough issue in connectionist AI, but mental representation of other minds presents a special challenge in at least two aspects: (1) Other minds are themselves capable of representing, leading to recursive and reflexive scenarios such as mind X representing a mind Y that itself represents mind X (see also Dennett, 1987). This point in particular complicates the connection with mirror neurons, which so far have not been observed to engage in recursive mirroring (although some have theorized about recursive functionality, see Gallese, 2007). (2) Other minds are invisible: we cannot see the minds of others, we can only guess at the existence of other minds via observation of behaviour. This point has di-

rect implications for learning about other minds: One might learn about another's behaviour via direct observation of that behaviour, but for learning about another's mind one needs forms of learning ability that incorporate inference from externally visible behaviour to (invisible) mental states. This sort of learning is difficult to capture with traditional AI conceptualizations of learning. Indeed, while computational work on ToM exists, the mechanisms for representing other minds are usually explicitly given and fixed (see e.g. Takano and Arita, 2006; Noble et al., 2010) (placing the focus on recursion depth instead). In this research we instead aim to let such mechanisms evolve from scratch, using a minimalistic evolutionary neural network model.

## Model

Agents are implemented as neural networks (NNs). Network architecture is evolved using a basic Genetic Algorithm. Agents interact in pairs. During its lifetime, each agent is part of multiple pairings. In each pair, there is a fixed role division: one agent acts at zero-order ToM ( $L_0$ ), meaning it ignores the other agent and simply reacts on basis of the state of the environment and its own mental state, and the other agent acts at first-order ToM ( $L_1$ ), meaning it tries to anticipate the behaviour of the  $L_0$  agent (at present, the  $L_1$  agent is simply tasked with predicting the  $L_0$  agent's behaviour). Each pair interacts for a set number of time-steps.

Our model is not intended to capture any specific social interaction scenario in particular. Instead we take a more abstract approach, in which the logic that determines the fitness payoff for performing a given action in a given state is generated randomly for each experiment (i.e. the fitness function is randomly generated for each run of the model). The idea is that if arbitrary fitness functions can be handled successfully, then the model has generality. Thus there is no concrete "task" to solve, there are merely *environmental states*, *mental states*, *actions*, and a randomly generated *base logic* that relates these elements.

*Environmental state*: bit-string of length  $N_e$  (set to 3 in the experiments discussed in this paper). The environmental state is shared between interacting agents (i.e. both agents see the same state). The environmental state changes every time-step. Each pair of agents sees each environmental state exactly once, in random order.



**Mental state:** bit-string of length  $N_m$  (set to 3 in the experiments discussed in this paper). Each agent has a private mental state, invisible to its interaction partner. Mental states remain constant over the course of the interaction of an agent pair.

**Action:** bit-string of length  $N_a$  (set to 3 in the experiments discussed in this paper). At each time-step, each agent outputs an action.

**Base logic:** generates the optimal  $L_0$  action choice for each (environmental state, mental state) pair. The base logic abstractly represents social scenarios. The base logic is implemented as a neural network, identical in kind to the NNs used for the agents, although with a few restrictions (we detail the way the base logic is generated after we explain the agent NN architecture below).

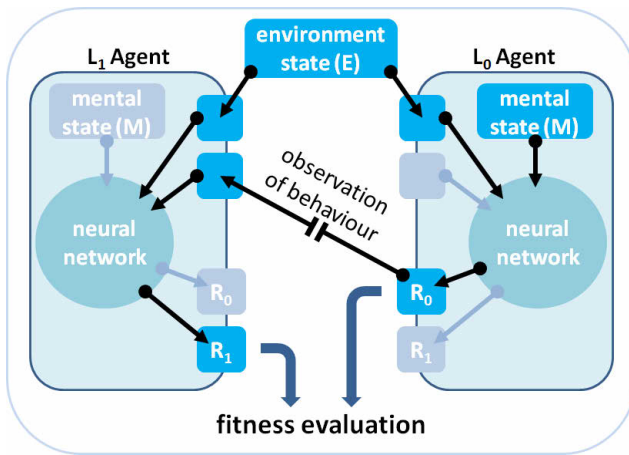


Figure 1. Schematic of agent interaction. The  $L_0$  agent computes its action ( $R_0$ ) from the (shared) environmental state and its (private) mental state. The  $L_1$  agent computes a prediction ( $R_1$ ) of this action. After the prediction is made,  $R_0$  is revealed to the  $L_1$  agent as data to drive its learning process.

Fitness scores for the action choices of the agent performing the  $L_0$  role are calculated as proximity to the optimal action as generated by the base logic, normalized to the  $[-1, +1]$  interval. Meanwhile, fitness scores for the action choices of the agent performing the  $L_1$  role are calculated as proximity to the action choice of the  $L_0$  agent. As such, the  $L_1$  agent must try to predict the action of the  $L_0$  agent, but the action choice of the  $L_0$  agent depends on the  $L_0$  agent's mental state, which is invisible to the  $L_1$  agent. Herein lies the challenge: in order for the  $L_1$  agent to be able to predict the  $L_0$  agent's future moves under future environmental states, the  $L_1$  agent must infer the  $L_0$  agent's mental state from the  $L_0$  agent's action choices under the current and preceding environmental states. By observing both the  $L_0$  agent's action choice and the environmental state that led the  $L_0$  agent to choose that action, the  $L_1$  agent has the necessary information to infer the  $L_0$  agent's mental state, and on the basis thereof it can predict the  $L_0$  agent's behaviour under other environmental states.

## Network species

We use a slightly unusual type of NN, which gives a central position to propagation order. A network consists of a list of neurons and a set of connections. Propagation simply follows the list order. The genome encodes for each connection the list-indices of the pre-synaptic neuron and the post-synaptic neuron. If the index of the post-synaptic neuron is smaller (or equal) to the index of the pre-synaptic neuron, then the connection runs against the propagation direction and is thus treated as a recurrent connection (meaning activation sent over it arrives at the next time-step). Otherwise, it runs along the propagation direction and is treated as a regular connection (meaning activation sent over it arrives at the same time-step). This approach avoids the trouble of deriving propagation order in free-form evolvable NN architectures, and facilitates later implementation of endogenously controlled propagation loops for recursive ToM.

The neuron list is composed of seven sections:  $2 \cdot N_m$  neurons for mental state input,  $2 \cdot N_e$  neurons for environmental state input,  $2 \cdot N_a$  neurons for partner action input,  $H$  hidden neurons,  $N_a$  neurons for  $L_0$  action output, again  $H$  hidden neurons, and finally  $N_a$  neurons for  $L_1$  action output. Connections between two neurons inside one and the same input or output section are not allowed. We provide two neurons for each input bit. Activation of one neuron signals a 1 value and activation of the other neuron signals a 0 value for the bit. These values can have very different implications, so we input them separately. When a NN acts as  $L_0$ , its responses are read from the first set of output neurons, and when it acts as  $L_1$ , the second set is read. When reading out responses at the output neurons, we translate neural activation values into binary values by converting negative values into zeroes and positive values into ones. Figure 2 shows the basic architecture (for  $N_m = N_e = N_a = 1$  and  $H = 2$ ). In the experiments discussed in this paper we used the following settings:  $N_m = N_e = N_a = 3$  and  $H = 16$ , making for a total of 56 neurons. Given that connectivity is evolved, it is very well possible for neurons to not be included in the circuitry, meaning that the "effective" net size will generally be smaller than 56. The maximum number of connections is limited to 100.

A neuron's activation is computed from the activation of the neurons that project to it, using a slight modification of the standard hyperbolic tangent activation function.

Activation function:

$$A_j = N_j \cdot \tanh \left[ 0.5 \cdot \sum (W_{ij}^A \cdot A_i) \right] + b_j$$

Where  $A_i$  is activation at neuron  $i$ ,  $W_{ij}^A$  is the weight of an activatory connection from  $i$  to  $j$ ,  $b_j$  is the (genetically encoded) activation bias for neuron  $j$ , and  $N_j$  is the neurotransmitter value at neuron  $j$ .  $N_j$  defaults to 1, but can be lowered if the neuron has any incoming *neurotransmitter connections*. Activation received over such connections is added to the  $N_j$  value.  $N_j$  is clipped to the  $[0, 1]$  range before the activation function is applied, and reset to 1 after propagation. This neurotransmitter logic is included to provide a simple mechanism for blocking signal transmission, which can simplify some neural computations (e.g. xor without hidden neurons). While theoretically speaking this does not expand the functionality



of the network species, we do find inclusion of such a neurotransmitter system to improve evolvability.

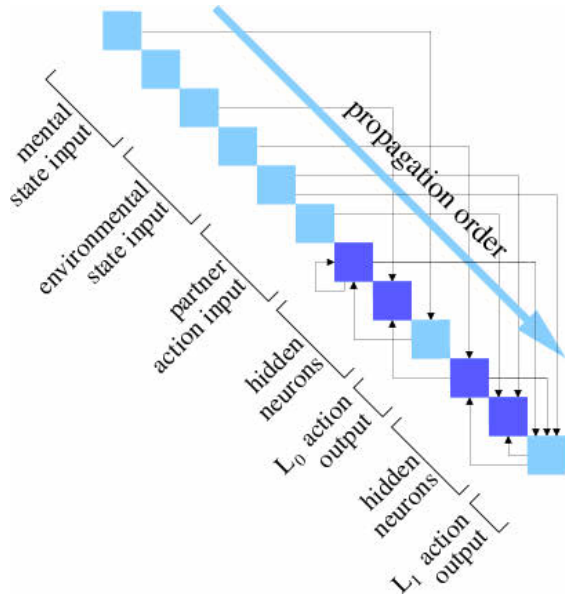


Figure 2. Neural network architecture concept. Networks consist of seven sections, with the size of each section determined by model parameters (see text). Propagation order is fixed. Connections running against the propagation order are treated as recurrent connections.

Evolution of learning ability is made possible using neuromodulation (Soltoggio et al., 2008). Similar techniques have previously been employed to evolve spatial representation ability in NNs (Arnold et al., 2012, 2013), so it stands to reason that it might allow for evolution of social representation ability as well. The basic idea is to introduce a special connection type that lets neurons send modulatory signals to one another, and to let these signals control connection weight change. This allows for evolution to shape the weight update dynamics of the networks by shaping the modulatory connectivity, which provides a basis for endogenously controlled behaviour change, i.e. a basis for learning ability. So in addition to their activation value, neurons have a modulation value, and in addition to standard activatory connections, there are modulatory connections. If neuron  $i$  has a modulatory connection to neuron  $j$ , then activation at  $i$  leads to modulation at  $j$ . Neurons' modulation values are computed in similar fashion to their activation values, but without involvement of neurotransmitter values.

Neuromodulation function:

$$M_j = \tanh \left[ 0.5 \cdot \sum (W_{ij}^M \cdot A_i) \right]$$

Where  $M_i$  is modulation at neuron  $i$ , and  $W_{ij}^M$  is the weight of a modulatory connection from  $i$  to  $j$ .

Weight updates are computed from the activation and modulation values at the pre- and post-synaptic neurons as follows:

$$\Delta W_{ij} = A_i^{g_{ij}^0} \cdot A_j^{g_{ij}^1} \cdot M_i^{g_{ij}^2} \cdot M_j^{g_{ij}^3}$$

Where  $\Delta W_{ij}$  is the change in the weight of the connection from neuron  $i$  to neuron  $j$ , and  $g_{ij}^0 \dots 3$  are binary genes that encode inclusion/exclusion of each term in the update function for this specific connection. Connection weights are clipped to  $[-1, +1]$ .

Weight updates are only performed when the network plays the role of an  $L_1$  agent (the behaviour target for  $L_0$  agents is static, so no learning is required there). When an agent plays the  $L_1$  role, propagation is performed twice. The first time, only the environmental state is given on the input neurons, propagation is performed, and the action prediction is read out on the  $L_1$  output neurons. Then the environmental state and the actual action choice of the  $L_0$  agent are given as input, propagation is performed, and connection weights are updated. In the second propagation round, output is ignored.

## Genetic Algorithm

Networks are evolved using a Genetic Algorithm, with mutation but no crossover. After each generation, agents are sorted by fitness, after which the worst performing two thirds of the population is replaced with copies of the best performing one third, to which then mutation is applied. Mutations can alter the following properties:

### Connections

- Pre- and post-synaptic neuron indices
- Type (activatory, neurotransmitter, modulatory)
- Update rule genes ( $g_{ij}^0 \dots 3$ )
- Existence

### Neurons

- Activation bias value

The "existence" property is used for addition and removal of connections (technically speaking there are always 100 connections in the network, but those with the existence property set to false are skipped over when the propagation logic is performed).

We additionally include some special mutation operators for modifying neural pathways, such as "inserting" a neuron into a pathway (given a connection from neuron  $x$  to neuron  $z$ , this operator picks a random neuron  $y$  and then replaces the original  $xz$  connection with a  $xy$  and a  $yz$  connection), and "removing" a neuron from a pathway (given a neuron  $y$ , it finds a connection projecting from some  $x$  to  $y$  and a connection from  $y$  to some  $z$ , and replaces the  $xy$  and  $yz$  connections with a single  $xz$  connection). Necessity of such operators has not been investigated here.

When mutating a network, we first pick a mutation rate using the following rule:

$$\begin{aligned} \text{rate} &= 16 \cdot R(0,1)^6 \\ \text{rate}_{\text{neuron}} &= \text{rate} / \text{neurons} \\ \text{rate}_{\text{connection}} &= \text{rate} / \text{connections} \end{aligned}$$

Where  $R(0,1)$  generates random numbers in the interval  $[0,1]$ , *neurons* is the number of neurons in the agent networks (56 in our experiments), and *connections* is the maximum number of connections in the agent networks (100 in our experiments). The resulting  $\text{rate}_{\text{neuron}}$  and  $\text{rate}_{\text{connection}}$  values are then used as the mutation probabilities per neuron and per connection, respectively. This slightly convoluted system for computing mutation rates is intended to help evolution by producing a good mixture of heavily and mildly mutated individuals, and was found to work better than fixed mutation rates.

### Generating the base logic

As noted above, we use random base logics instead of specific social scenarios. Base logics are implemented as special instances of the neural network architecture described above, and generated using the same genetic algorithm, just using a different fitness function than used when evolving agent NNs. The base logic serves as the target phenotype for  $L_0$  agents (i.e. fitness of  $L_0$  behaviour is judged as proximity to the base logic). As the base logic is just a special exemplar of the same neural network species, it takes environmental and mental states as input and returns actions as output. The fitness function used for generating base logics assesses the suitability of this input-output mapping as a social task, using the product of two values:

*Environmental state relevance*: for each possible mental state, we count the number of different action outputs that can be obtained by varying the environmental state. This gives an indication of the relevance of the environmental state for optimal action choice. If different environmental states do not lead to different optimal actions, then little can be learned by observing another agent in different environmental states. Thus the base logic should have a high value for environmental state relevance, to make learning *possible*.

*Mental state relevance*: for each possible environmental state, we count the number of different action outputs that can be obtained by varying the mental state. This gives an indication of the relevance of the mental state for optimal action choice. If different mental states do not lead to different optimal actions, then there is no need for  $L_1$  agents to learn about the mental state of the  $L_0$  agent. Thus the base logic should have a high value for mental state relevance as well, to make learning *necessary*.

Expressed in formulaic form:

$$\text{fitness}_{\text{logic}} = \sum_{\text{mst}}^{\text{m}} \text{actions}(\text{m}) \cdot \sum_{\text{env}}^{\text{e}} \text{actions}(\text{e})$$

Where *mst* is the set of possible mental states, *env* is the set of environmental states,  $\text{actions}(\text{m})$  is the number of distinct actions that can be obtained for mental state *m* by varying the environmental state, and  $\text{actions}(\text{e})$  is the number of distinct actions that can be obtained for environmental state *e* by varying the mental state.

While we use the same network architecture, some limitations are imposed when a network is used as base logic. First off, the base logic should remain constant over time-steps, so modulatory and recurrent connections are disabled. Secondly, the base logic only provides the optimal actions for  $L_0$  (for  $L_1$ , optimal action is imitation of  $L_0$ ), so neurons beyond the first

output section are omitted. Thirdly, to ensure that agents can viably replicate the base logic phenotype, we limit the number of neurons in the hidden neuron section to half of that used in the agent networks (i.e. 8 instead of 16).

We evolve a base logic population of 225 networks for 2000 generations, and retain the best individual of the final generation as the base logic for the experiment. Then a population of 225 agent networks is evolved for 2,000,000 generations.

### Evolving the agent population

In each generation, we split the population into 25 groups of 9 agents each. Within each group, every possible agent pairing interacts twice (once for each role assignment). At the start of each interaction, the  $L_0$  agent generates a mental state, which remains constant throughout the interaction. Then the  $L_0$  agent is exposed to every possible environmental state, in random order, while the  $L_1$  agent tries to predict the  $L_0$  agent's actions, seeing the shared environmental state but not the  $L_0$  agent's mental state. After each action of the  $L_0$  agent, the actual action choice is revealed to the  $L_1$  agent (i.e. fed into its "partner action" input neurons), and weight update logic is performed. Each environmental state is seen only once per interaction, so simply remembering the partner's action choice is no viable strategy. For the  $L_0$  agent, fitness payoff for an action is simply proportional to the action's proximity to the optimal action as given by the base logic. Below we use the performance of the  $L_1$  agent on the last time-step of the interaction as a measure of the success of the learning process. However,  $L_1$  fitness as used by the genetic algorithm is measured over all steps, so that a faster learner will have better fitness than a slow learner even if they perform equally well on the last time-step of the interaction.

At the end of each generation, agents are ranked per group, on basis of their  $L_0$  performance and  $L_1$  performance, with  $L_0$  performance taking precedence over  $L_1$  performance. That is, if agent *X* has a higher  $L_0$  performance than agent *Y*, then *X* will be ranked above *Y*, independent of the  $L_1$  performance scores. When *X* and *Y* have identical  $L_0$  scores (a very common occurrence, especially once optimality on the  $L_0$  task has been achieved), rank is decided by the  $L_1$  score. This way of ranking avoids a trap in social evolution. If evolution co-opts the neural circuitry that determines  $L_0$  behaviour for prediction of other agents'  $L_0$  behaviour (i.e.  $L_1$  behaviour), then when a mutant with better circuitry for  $L_0$  behaviour appears, this mutant will have *worse* prediction ability with respect to the behaviour of its non-mutant peers (and those peers will have worse prediction ability with respect to the mutant). To prevent such effects from obstructing evolution of  $L_0$  behaviour,  $L_0$  performance should take precedence of  $L_1$  behaviour. The best one third of each group (3 agents with the settings used here) overwrites the remaining two thirds with copies of themselves (so 2 copies per agent), to which mutation is applied. So effectively the parent agents each have 3 offspring, 2 mutated and 1 unmutated.

## Results

Eight trials of the experiment described above were performed. Table 1 shows performance results for the final 1000 generations of each trial, and Figure 3 shows the evolution

process of a representative run. The scores displayed for  $L_0$  are computed over individuals that have been retained without mutation from the previous generation (i.e. the unmutated offspring of the previous generation). Note that this implies that inclusion of an individual in the performance score is decided *before* its performance is measured. Such scores provide a good performance measure, as they are neither distorted much by mutation (as whole population averages are) nor by luck (as population best scores are). Scores displayed for  $L_1$  are computed over all pairings between such individuals.

Run #	Max $L_1$ score without learning	$L_0$ score	$L_1$ score
1	.21	.9195	.9841
2	0	.9984	.9062
3	.20	.9861	.9641
4	.05	.9506	.8972
5	.34	.9199	.9745
6	0	.9961	.9690
7	.21	.9995	.9941
8	.29	.8839	.8974
average	.1625	.9568	.9483

Table 1. Performance scores. Scores are averages over the last 1000 generations of each run. Score averages for  $L_0$  are taken over all individuals that have been copied from the previous generation without mutation. Score averages for  $L_1$  are taken over all pairings between such individuals.

Variation in partner  $L_0$  behaviour will harm  $L_1$  performance of even optimal agents. As such, perfect performance cannot be expected. To allow for assessment of the evolved learning ability, we calculated the maximal expected fitness score agents without learning could obtain for each run's base logic, and show these alongside the performance scores in Table 1. These are the expected  $L_1$  scores achieved by a hypothetical non-learning agent that for each environmental state simply picks the action that (over all possible mental states for an optimal  $L_0$  partner agent) yields the best expected score. Note that the expected score for random behaviour is 0. We can see that  $L_1$  performance in all runs widely exceeds the computed maxima for non-learning agents, showing that learning ability was indeed evolved. Average performance over all runs is well over 90% of the theoretical maximum, and some runs get very close to maximum performance (runs 3, 6 & 7).

These results indicate that the model, while far from perfect, is capable of producing agents that can learn how their interaction partner maps environmental states to actions. In that mapping, the partner's mental state plays a central role. As such, it seems that by observing their partner's behaviour, the agents in some form or another get a grasp on their partner's

mental state. This suggests that these agents have evolved a primitive form of first-order Theory of Mind.

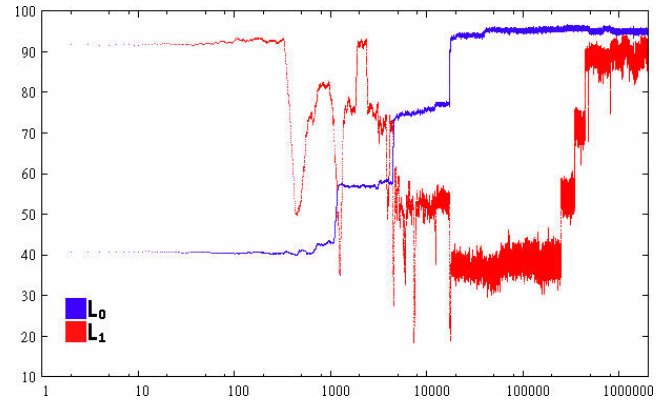


Figure 3. Evolution process of example run (run 4 in Table 1). As in Table 1,  $L_0$  scores are averages over unmutated individuals and  $L_1$  scores are averages over interactions between such individuals. Data points are smoothed over 100 generations. X-axis in log-scale. Initially the population adopted a simple  $L_0$  behaviour, easy to imitate (hence high  $L_1$  scores early on) but low in fitness. As the  $L_0$  behaviour improves we see the  $L_1$  performance fall, and then climb back up again as the learning ability necessary for prediction of the more complex  $L_0$  behaviour evolves. Eventually the system settles in a state with high performance for both  $L_0$  and  $L_1$ .

## Future Work

The networks evolved here act as though they have first order ToM, but we have yet to establish how the mental state of partner agents is represented in the networks' activation patterns and/or weight modifications. Our primary future goal is to investigate this. It is well known that more often than not, networks evolved or trained to solve a given task represent their knowledge in highly diffuse and distributed fashion. However, as we have shown elsewhere (Arnold et al., 2012, 2013), when evolution and learning are combined, interactions between them tend to give rise to more organized forms of representation. In the present work too, we have a combination of learning and evolution at work.

Secondly, specific to ToM, we will investigate whether, in predicting an  $L_0$  partner's behaviour, these nets use the circuitry they use when they themselves are performing at  $L_0$ . This would constitute a mirror-neuron-like, "placing oneself in another's shoes" approach to the problem.

Beyond the above, we aim to extend this research in the following directions: 1) More complex scenarios for the  $L_1$  response (i.e. not mere prediction, but acting in anticipation of the  $L_0$  agent's action). 2) Extension to higher (recursive) orders of ToM (by introducing endogenously controlled propagation looping in the NN architecture). 3) Once the model works for mental and environmental states of sufficient size, replacement of the randomly generated base logic with simple games or cognitive psychology experiments that involve ToM.

## References

- Arnold, S., Suzuki R. and Arita, T. (2012). Second Order Learning and the Evolution of Mental Representation. In *Artificial Life XIII: Proceedings of the Tenth International Conference on Artificial Life*, pp. 301–308. MIT press, Cambridge, MA.
- Arnold, S., Suzuki, R., and Arita, T. (2013). Selection for Reinforcement-Free Learning Ability as an Organizing Factor in the Evolution of Cognition. *Advances in Artificial Intelligence*, vol. 2013, Article ID 841646, 13 pages.
- Dennett, D. C. (1987). *The Intentional Stance*. MIT Press / Bradford Books, Cambridge, MA.
- Gallese, V. (2007). Before and below ‘theory of mind’: embodied simulation and the neural correlates of social cognition. *Phil. Trans. R. Soc. B* 362, 659–669. doi:10.1098/rstb.2006.2002
- Noble, J., Hebborn, T., Van Der Horst, J., Mills, R., Powers, S. T. and Watson, R. (2010). Selection pressures for a theory-of-mind faculty in artificial agents. In *Artificial Life XII: Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pp. 615–615. MIT Press, Cambridge, MA.
- Premack, D. G., Woodruff, G. (1978). Does the chimpanzee have a theory of mind?. *Behavioral and Brain Sciences* 1 (4): 515–526. doi:10.1017/S0140525X00076512.
- Soltoggio, A., Bullinaria, J. A., Mattiussi, C., Dürr, P. and Floreano, D. (2008). Evolutionary Advantages of Neuro-modulated Plasticity in Dynamic, Reward-based Scenarios. In *Artificial Life XI: Proceedings of the Tenth International Conference on Artificial Life*, pp. 569–576. MIT Press, Cambridge, MA.
- Takano, R. and Arita, T. (2006). Asymmetry between even and odd levels of recursion in a theory of mind. In *Artificial Life X: Proceedings of the Tenth International Conference on Artificial Life*, pp 405–411. MIT Press, Cambridge, MA.



# Coevolutionary Cartesian Genetic Programming in FPGA

Radek Hrbáček and Michaela Šikulová

Brno University of Technology, Faculty of Information Technology,  
Božetěchova 2, 612 66 Brno, Czech Republic  
xhrbac01@stud.fit.vutbr.cz, isikulova@fit.vutbr.cz

## Abstract

In this paper, a hardware platform for coevolutionary cartesian genetic programming is proposed. The proposed two-population coevolutionary algorithm involves the implementation of search algorithms in two MicroBlaze soft processors (one for each population) interconnected by the AXI bus in Xilinx Virtex 6 FPGA. Candidate programs are evaluated in a domain-specific virtual reconfigurable circuit incorporated into custom MicroBlaze peripheral. Experimental results in the task of evolutionary image filter design show that we can achieve significant speed-up (up to 58) in comparison with highly optimized software implementation.

## Introduction

Cartesian genetic programming (CGP) – a special variant of genetic programming (GP) – has been successfully applied to a number of challenging real-world problem domains (Miller, 2011). However, the computational power that evolutionary design based on CGP (as well as on standard GP) needs for obtaining innovative results is enormous for most applications. Often, the fitness in GP is calculated over a set of *fitness cases* (Vanneschi and Poli, 2012). A fitness case corresponds to a representative situation in which the ability of a program to solve a problem can be evaluated. Fitness case consists of potential program inputs and target values expected from a perfect solution as a response for these program inputs.

A set of fitness cases is typically a small sample of the entire domain space. The choice of how many fitness cases (and which ones) to use is often crucial since whether or not an evolved solution will generalize over the entire domain depends on this choice. However, in the case of digital circuit evolution, it is necessary to verify whether a candidate  $n$ -input circuit generates correct responses for all possible fitness cases (input combinations, i.e.  $2^n$  assignments). It was shown that testing just a subset of  $2^n$  fitness cases does not lead to correctly working circuits (Imamura et al., 2000). Recent work has indicated that this problem can partially be eliminated in real-world applications by applying formal verification techniques (Vasicek and Sekanina, 2011).

Hillis (1990) introduced an approach that can automatically evolve subsets of fitness cases concurrently with problem solution. Hillis used a two-population coevolutionary algorithm (CoEA) applied to a test-based problem in the task of minimal sorting network design. Subsets of test cases used to evaluate sorting networks evolved simultaneously with the sorting networks. Evolved sorting networks were used to evaluate the test cases subsets. The fitness of each sorting network was measured by its ability to correctly solve fitness cases while the fitness of the fitness cases subsets was better for those that could not be solved well by currently evolved sorting networks.

Coevolutionary algorithms are traditionally used to evolve interactive behavior which is difficult to evolve with an absolute fitness function. The state of the art of coevolutionary algorithms has recently been summarized in (Popovici et al., 2012). A *test-based problem* is defined as a co-search or co-optimization problem with two populations – population of candidate solutions and population of *tests* (subsets of the fitness cases set).

In our previous work, inspired by *coevolution of fitness predictors* (Schmidt and Lipson, 2008) and the principles of the *competitive coevolution* introduced by Hillis (1990), we proposed a two-population coevolutionary CGP algorithm running on an ordinary processor in order to accelerate the task of symbolic regression (Šikulová and Sekanina, 2012b) and the evolutionary image filter design (Šikulová and Sekanina, 2012a). For our benchmark problems (5 symbolic regression problems and salt-and-pepper noise filter design) we have shown that the (median) execution time can be reduced 2-5 times in comparison with the standard CGP.

Despite the acceleration based on fitness cases coevolution, the CGP design is still computationally very intensive design method. Therefore an FPGA based acceleration platform has been designed. Modern FPGAs provide cheap, flexible and powerful platform, often outperforming common workstations or even clusters of workstations in particular applications. Vasicek and Sekanina (2010) introduced a new FPGA accelerator of CGP with the aim to provide both high performance and low power. The architecture

contains multiple instances of *virtual reconfigurable circuit* (VRC, Sekanina (2003)) to evaluate several candidate solutions in parallel.

Inspired by the FPGA accelerator of CGP, we propose a hardware platform for parallel two-population CoEA and show that by using this platform, the execution time of evolutionary design using CGP can be significantly reduced. The proposed hardware accelerated coevolutionary CGP is compared with hardware-accelerated standard CGP and with a highly optimized software implementation of coevolutionary CGP in the task of evolutionary image filter design.

The paper is organized as follows. The next section introduces the idea of coevolution in cartesian genetic programming. In the following section the architecture of the proposed accelerator is presented. The remaining section is devoted to experimental evaluation of the accelerator in the benchmark problem – the image filter evolution. Conclusions are given in the last section.

### Coevolution in Cartesian Genetic Programming

In standard CGP (Miller, 2011), a candidate program is represented in the form of directed acyclic graph, which is modelled as an array of  $n_c \times n_r$  (columns  $\times$  rows) programmable elements (nodes). The number of primary inputs,  $n_i$ , and outputs,  $n_o$ , of the program is defined for a particular task. Each node input can be connected either to the output of a node placed in previous  $l$  columns or to one

of the program inputs. The  $l$ -back parameter, in fact, defines the level of connectivity and thus reduces/extends the search space. Feedback is not allowed. Each node is programmed to perform one of  $n_a$ -input functions defined in the set  $\Gamma$ . Each node is encoded using  $n_a + 1$  integers where values  $1 \dots n_a$  are the indexes of the input connections and the last value is the function code. Every individual is encoded using  $n_c \cdot n_r \cdot (n_a + 1) + n_o$  integers.

A simple  $(1+\lambda)$  evolutionary algorithm is used as a search mechanism. It means that CGP operates with the population of  $1 + \lambda$  individuals (typically,  $\lambda$  is between 1 and 20). The initial population is constructed either randomly or by a heuristic procedure. Every new population consists of the best individual of the previous population (so-called parent) and its  $\lambda$  offspring. In each generation, an offspring with equal or better fitness than the parent's is chosen as the new parent. The offspring individuals are created using a point mutation operator which modifies up to  $h$  randomly selected genes of the chromosome, where  $h$  is a user-defined value. The algorithm is terminated when the maximum number of generations is exhausted or a sufficiently working solution is obtained.

There are two concurrently evolving populations in the proposed coevolutionary algorithm: (1) candidate programs evolving using CGP and (2) tests (*fitness cases subsets*, *abb. FCSs*) evolving using a simple genetic algorithm. Both populations evolve simultaneously and interact through the fitness function.

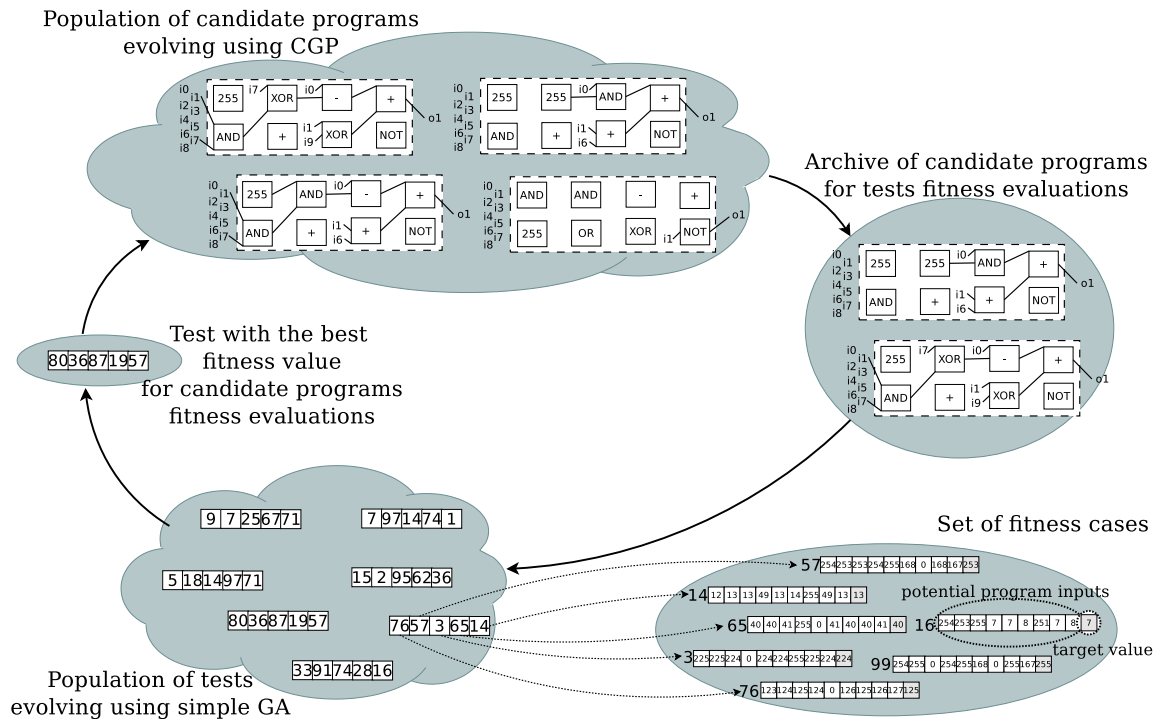


Figure 1: Populations in coevolutionary CGP – candidate programs and tests.

Test is a subset of the fitness cases set, therefore every test is encoded as a fixed-sized array of pointers to elements in the fitness cases set. In addition to one-point crossover and mutation, a randomly generated tests replacing the worst-scored tests in each generation has been used.

The aim of coevolving tests and candidate programs is to allow both candidate programs and tests to enhance each other automatically until a satisfactory problem solution has been found. Figure 1 shows the overall scheme of the proposed method. If the top-ranked candidate program fitness value (in the actual generation of candidate programs evolution) has changed against the previous generation, the top-ranked candidate program is copied to the archive of candidate programs. The archive of candidate programs is a circular list that is used for tests evaluation. Tests (in the tests evolution) are evaluated using candidate programs from the archive as follows. Each candidate program from the archive is executed for all fitness cases in the test. The test with the worst mean fitness value for candidate programs from the archive is selected as the top-ranked test in the actual generation. This test is then used to evaluate candidate programs in the candidate programs evolution. This fitness interaction approach allows to improve candidate programs using the fitness cases, which cannot be correctly solved by currently evolved candidate programs yet.

### Hardware platform design

The evolutionary design includes two basic steps alternating in each generation – generation of new population and evaluation. Since the evaluation step consists in multiple running or simulating of candidate program and computing chosen fitness, a significant acceleration can be achieved by means of task or data parallelism, while the best throughput can be achieved using custom hardware.

On the contrary, the evolutionary process control is, by its nature, suitable rather for running on a universal processor, moreover in the case of CoEAs two evolutionary processes need to be executed in parallel with the ability to communicate with each other.

These requirements have been taken into account when choosing the target platform. Currently, two suitable alternatives are available – conventional FPGAs and a combination of a processor and programmable logic (e.g. Xilinx Zynq All Programmable SoC, Dobai and Sekanina (2013)). Table 1 compares several devices available in our institution as part of a development kit with respect to the configurable logic

Table 1: Target platforms comparison.

	device	logic cells	block RAM
	Virtex 6 XC6VLX240T	241,152	14,976 Kb
	Virtex 7 XC7K325T	326,080	16,020 Kb
	Zynq 7020 XC7Z020	85,000	4,480 Kb

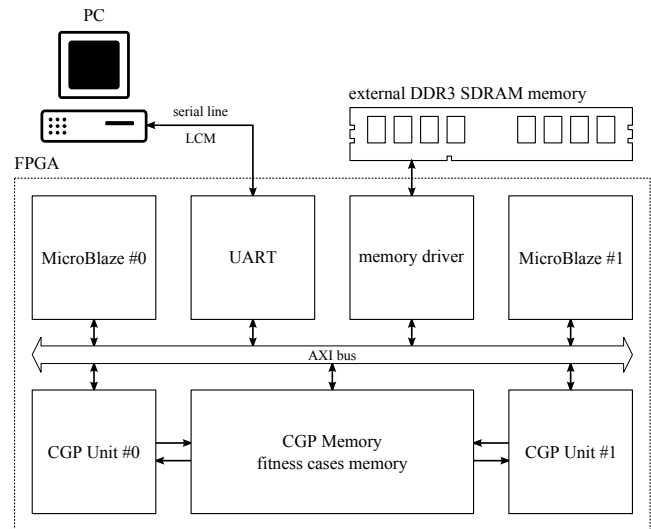


Figure 2: Hardware platform architecture.

cells count and the amount of block RAM. It is obvious, that the Zynq platform offers much less flexibility in terms of custom logic comparing to Virtex FPGA family. Therefore, a standard FPGA has been chosen as a more flexible option.

Despite the fact that standard FPGAs do not have hard processors, wide choice of soft processors under various licences are available. The most suitable choice for Xilinx devices is the MicroBlaze soft processor, offering sufficient performance while occupying a reasonable area. Figure 2 shows the proposed hardware platform architecture. The system consists of two MicroBlaze soft processors supplemented by two independent acceleration units (CGP Units) and fitness cases memory (CGP Memory). All components are interconnected by the AXI bus and additional memory channels are introduced for fitness cases transfers. Com-

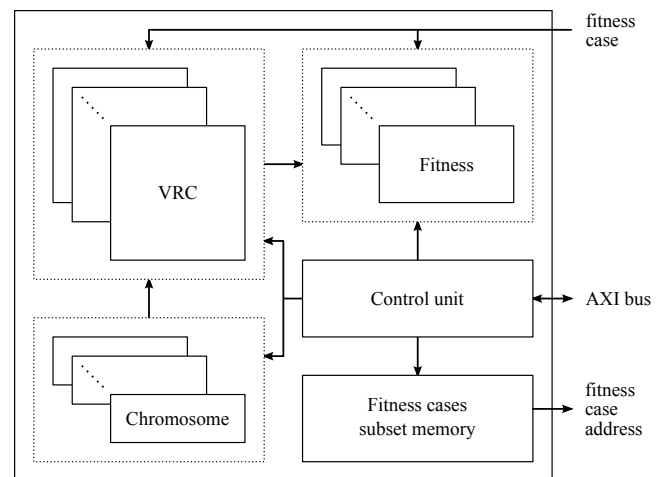


Figure 3: Detailed architecture of CGP Unit.

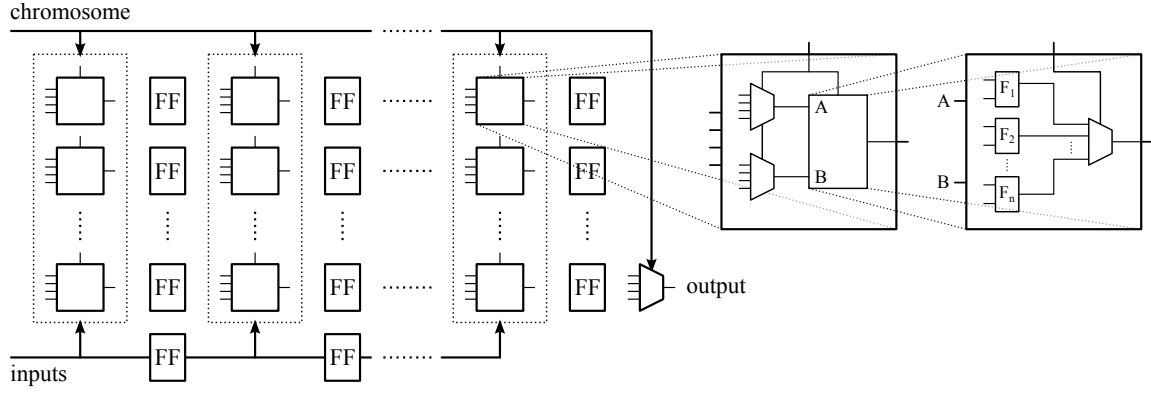


Figure 4: Virtual Reconfigurable Circuit (VRC).

munication with a service application running on a PC is performed through serial port (UART) and LCM communication library<sup>1</sup>. The dual MicroBlaze system utilizes AXI Mailbox component, which enables to pass simple messages between the processors (control and status messages, chromosomes, fitness values etc.).

CGP Unit (Figure 3) includes a set of subcomponents, each composed of one virtual reconfigurable circuit (VRC), fitness unit and chromosome register. Moreover, the CGP Unit includes a common control unit and FCS memory, each subcomponent is fed with the same data. The control unit is responsible for the communication between the MicroBlaze processor and the peripheral and for controlling the fitness computation. There are several configuration and status registers that are memory mapped on the AXI bus together with the test memory. By setting a specific bit in the control register, the fitness computation starts. Fitness cases are addressed indirectly using the test memory, which is addressed sequentially by the control unit. In the case of image filter design, each fitness case consists of chosen, e.g.  $3 \times 3$  or  $5 \times 5$ , pixel neighbourhood from the noisy image and one pixel from the original image. The noisy pixels are processed in the VRCs and together with the clean pixel, properly delayed, come to the fitness unit. After a specified number of fitness cases is processed, the control unit saves the current fitness values and notifies the MicroBlaze processor by changing the status register value.

The VRC architecture is shown in Figure 4. According to the program representation in CGP, the VRC comprises a grid of nodes, called configurable function blocks (CFBs), interconnected in such a way that each block can access all other blocks in previous columns and the VRC inputs. Both VRC's inputs and CFB's outputs are registered and delayed, so that the VRC is fully pipelined while keeping the  $l$ -back parameter of arbitrary choice. Thanks to the pipelining, the

<sup>1</sup>Lightweight Communications and Marshalling (LCM) is a set of libraries and tools for message passing and data marshalling originally designed by the MIT DARPA Urban Challenge Team.

Table 2: Functions implemented in CFBs according to Sekanina et al. (2011).

#	function	#	function
0	255	8	$i_1 \gg 1$
1	$\frac{i_1}{i_2}$	9	$i_1 \gg 2$
2	$\frac{i_2}{i_1}$	10	$(i_1 \ll 4) \vee (i_2 \gg 4)$
3	$\frac{i_1}{i_2} \vee i_2$	11	$i_1 + i_2$
4	$\frac{i_1}{i_2} \wedge i_2$	12	$i_1 +^s i_2$
5	$\frac{i_1}{i_2} \wedge i_2$	13	$(i_1 + i_2) \gg 1$
6	$\frac{i_1}{i_2} \wedge i_2$	14	$\max(i_1, i_2)$
7	$i_1 \oplus i_2$	15	$\min(i_1, i_2)$

VRC is able to process one fitness case per clock cycle.

Each CFB has the same structure. The input data are selected using two multiplexers and forwarded to several functions (functions used for image filter design are listed in Table 2), the output value is selected by an output multiplexer. The configuration of the multiplexers is determined by specific genes of the chromosome.

The output of each VRC is connected to separate fitness unit (Figure 7). Two different fitness functions are computed simultaneously – squared and absolute error:

$$f_{sq} = \sum_{i=1}^N (x_i - y_i)^2, \quad (1)$$

$$f_{abs} = \sum_{i=1}^N |x_i - y_i|,$$

where  $x_i$  is the clean pixel,  $y_i$  the VRC output corresponding to the  $i$ -th fitness case and  $N$  the number of fitness cases. These fitness functions are very similar to the MSE and MDPP functions (commonly used for image filter design), except for normalization with the number of pixels  $N$ . Since division is a very demanding operation, its removal saves a lot of resources without any impact on the application in EAs. While performing experiments, one can choose which



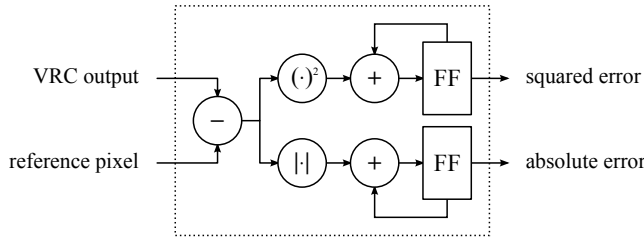


Figure 7: Fitness unit.

fitness function is used for the evaluation.

The CGP Memory component (Figure 8) is designed to achieve very high throughput. One write port (connected to the AXI bus) and two read ports enable to supply both CGP Units with data. These ports have different data widths because the AXI bus is 32 bit wide, but the width of the fitness case depends on the chosen pixel neighbourhood (80 bits for  $3 \times 3$ , 208 bits for  $5 \times 5$ ). Therefore the memory has to be divided into 8 bit wide blocks and the read and write ports have to be treated in a different way. The fitness case (data output of the read port) is a concatenation of values from all these blocks from the same address. When writing from the AXI bus side, at most 4 blocks are updated at the same time. The total memory size depends on the chosen pixel neighbourhood and the maximum training image size we want to use. In our design, the number of fitness cases is limited to 65,536 due to fixed address width (16 bit), then the maximum memory capacity is  $80 \cdot 65,536 \approx 5,243$  Kb for  $3 \times 3$ , respectively  $208 \cdot 65,536 \approx 13,632$  Kb for  $5 \times 5$  neighbourhood. Note that these sizes still fit into the Virtex devices, but not into the Zynq SoC (see Table 1).

Thanks to these hardware components, the fitness calculation is very efficient. The remaining steps of the evolutionary process (individuals manipulation, communication) take place on the MicroBlaze processors.

The evolutionary design is running as follows. At the beginning, original and noisy images are transferred to the external DDR3 memory, fitness cases are put together and copied to the CGP Memory. After that, the design process is initiated. Timing diagram in Figure 5 shows the steps of a single generation. The population is divided into  $N_{ch}$  chunks of  $P_{ch}$  individuals depending on the number of

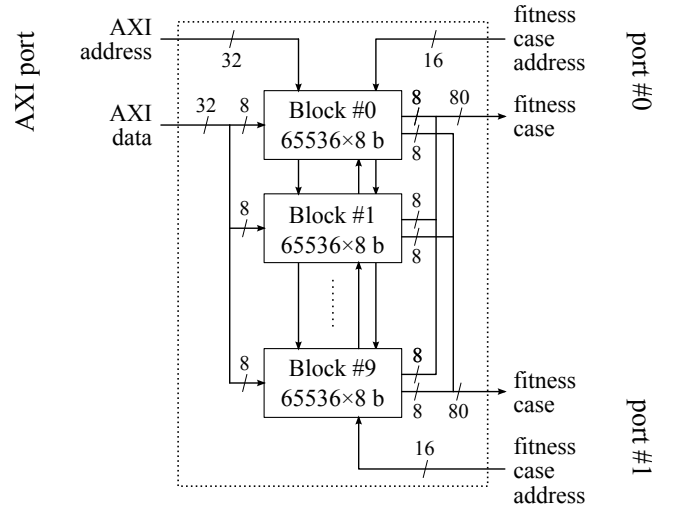


Figure 8: Architecture of CGP Memory.

VRCs  $N_{VRC}$  and the population size  $P$ :

$$N_{ch} = \left\lceil \frac{P}{N_{VRC}} \right\rceil, \quad P_{ch} = \left\lceil \frac{P}{N_{ch}} \right\rceil. \quad (2)$$

In each generation, individuals belonging to the first chunk need to be mutated and transferred to the CGP Unit before the fitness computation is executed. For every succeeding chunk (except for the last one including the last individuals of the population), the mutations and chromosome transfers can be overlapped with the fitness computation (all chromosome registers are shadowed). To achieve the best hardware utilization, the fitness computation time  $t_f$  has to be longer than the time  $t_m$  spent on the mutations and transfers. Ignoring some overhead, the total time per generation  $t_g$  is than:

$$t_g = t_m + (N_{ch} - 1) \cdot \max(t_m, t_f) + t_f. \quad (3)$$

Finally, when the evolution is completed, the best individual's chromosome is sent to the PC.

The coevolutionary design process is slightly more difficult, as it can be seen in Figure 6. The image filter evolution is running almost the same way except for the fitness cases subset, which is being evolved in parallel. For the purpose of FCSs evaluation, the best evolved filters are saved to an archive of candidate filters. The FCSs evolutionary process

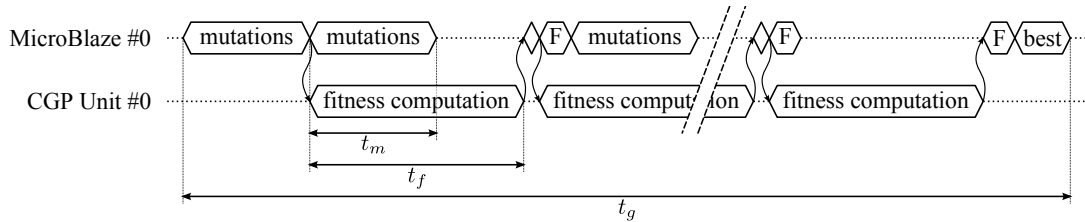


Figure 5: Timing diagram of the evolutionary process.

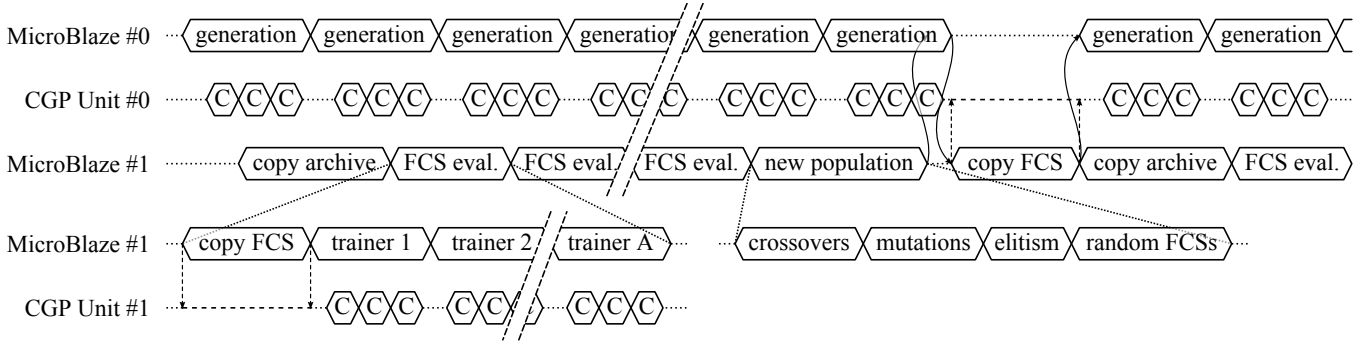


Figure 6: Timing diagram of the coevolutionary process.

is based on a simple genetic algorithm, the FCS chromosome is represented by a fixed-sized array of integers. In each generation, all FCS individuals are evaluated using all filters from the archive (trainers), the fitness value of the  $i$ -th individual is the mean value of the particular fitnesses:

$$f_{\text{FCS}}^i = \frac{1}{A} \sum_{j=1}^A f(i, j), \quad (4)$$

where  $A$  is the archive size and  $f(i, j)$  is the fitness (either squared error  $f_{\text{sq}}$  or absolute error  $f_{\text{abs}}$ ) of the  $j$ -th trainer on the  $i$ -th FCS. After all FCSs are evaluated, new population is created using standard genetic operators. Specified number of individuals is obtained by one-point crossovers and the new individuals are mutated with some probability. In order to exert the selective pressure, elitism is introduced by keeping the best individual unchanged and making a few mutated clones. Finally, the rest of the population is generated randomly to preserve genetic variability. At the end of each generation, the filter evolutionary process is notified and at the right moment (after finishing the entire generation), the FCS is copied to the CGP Unit #0. No FCSs sharing between MicroBlaze processors is required.

## Experimental results

This section presents benchmark problems, experimental setup and experimental evaluation of the proposed hardware accelerated approach and its comparison with the software approach.

In order to evaluate the proposed approach, salt-and-pepper noise filters were designed using standard CGP and coevolutionary CGP. This type of noise is characterized by noisy pixels with the value of either 0 or 255 (for 8-bit gray-scaled images). The Lena training image with size  $256 \times 256$  pixels was corrupted by 5%, 10%, 15% and 20% salt-and-pepper noise. The evolved filters were tested on 14 different images (Gonzalez et al., 2009) containing the same type of noise.

CGP was used according to Sekanina et al. (2011), i.e.  $n_c = 8$ ,  $n_r = 4$ ,  $l = 7$ ,  $n_i = 9$ ,  $n_o = 1$ ,  $\lambda = 19$ , every node had two inputs, the number of mutations per new individual was  $h = 5$  and  $\Gamma$  contained the functions from Table 2. The archive of candidate programs had capacity of 20 elements.

FCSs were evolved using a simple GA, where 3-tournament selection, single point crossover and mutation up to 2 % of chromosome were used. Elitism and random individuals were used to exert selective pressure and preserve genetic variability. For the GA, various chromosome lengths were tested, particularly, 1.5625 %, 3.125 %, 6.25 %, 12.5 %, 25 % and 50 % of total number of fitness cases in the training set. For each FCS size, 100 independent runs were performed and the evolution/coevolution was terminated after 100,000 generations of CGP.

The proposed coevolutionary algorithm accelerated using FPGA was compared with the standard CGP algorithm in terms of filtering quality of evolved filters and with the highly optimized coevolution implementation running on an ordinary processor in terms of the execution time.

The quality of filtering was expressed using a measure typically used in the image processing community – as a peak signal-to-noise ratio (PSNR):

$$\text{PSNR}(x, y) = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i,j} (x(i, j) - y(i, j))^2}, \quad (5)$$

where  $M \times N$  is the size of the image,  $x$  denotes the original image,  $y$  the filtered image and  $i, j$  are indexes of a pixel in the image. Figure 9 shows that using coevolutionary CGP running on an FPGA we are able to evolve image filters of comparable (or better) quality than standard CGP for all noise intensities. Furthermore, the higher the noise intensity, the smaller fitness cases subset can be used to get acceptable results.

Software and hardware performance comparison for standard CGP can be found in Table 3. The software implementation is a command line tool written in C++ utilizing OpenMP library for running in multiple threads and SSE 4.1 instruc-

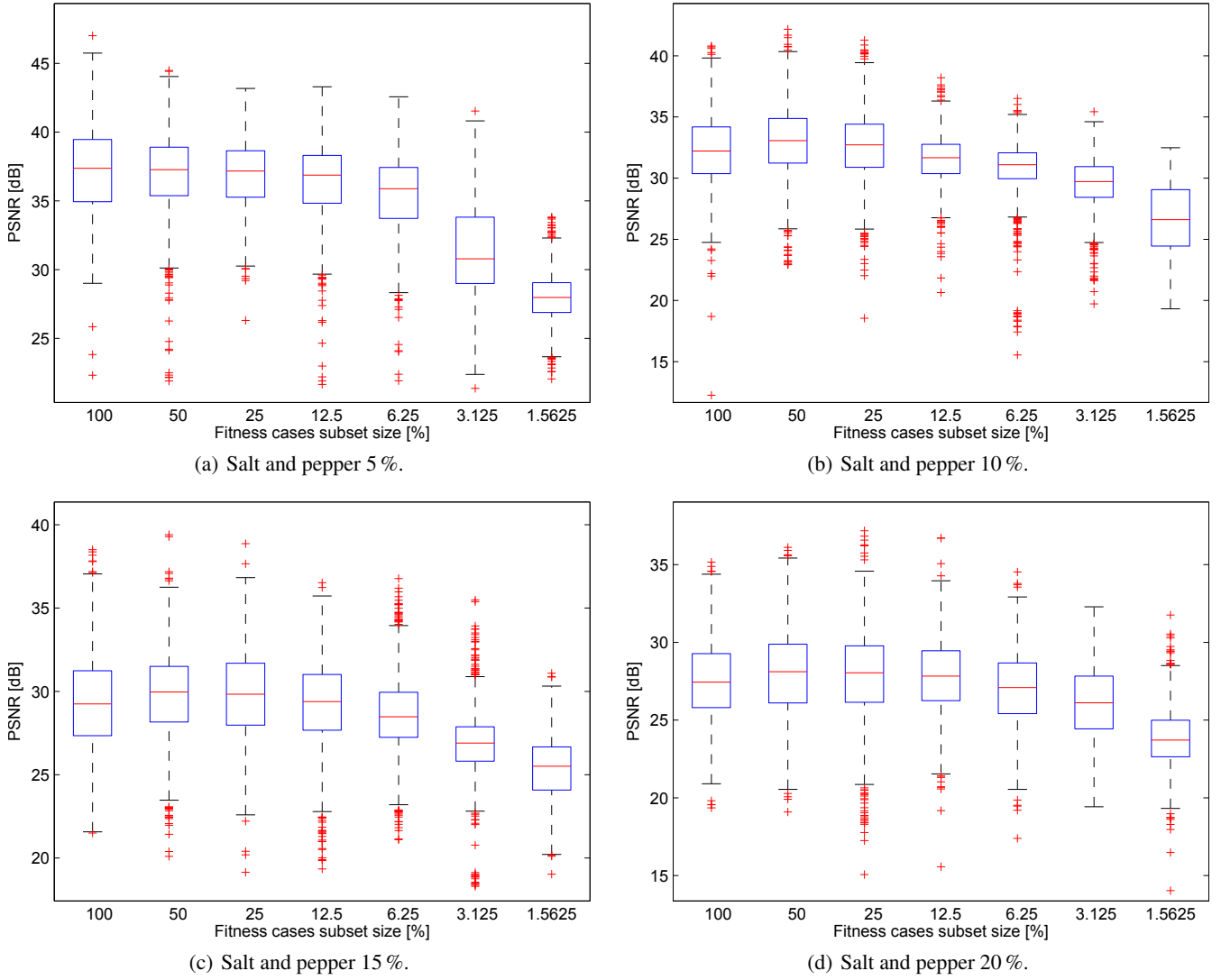


Figure 9: PSNR statistics calculated from 100 evolved filters (100 independent runs using the Lena training image for each noise intensity and each FCS size) for the 14 test images.

tions enabling to process 16 fitness cases in a single step. Before evaluation, each chromosome is analyzed to exclude the inactive nodes. The performance tests were performed on the Intel Core i7-860 processor (2.8 GHz), allowing 8 threads to be running simultaneously. The hardware platform configuration was as follows: 7 VRCs in the CGP Unit #0, 6 VRCs in the CGP Unit #1, in total  $N_{VRC} = 13$ , the entire system was running on 100 MHz frequency. Despite a very efficient software implementation and powerful processor, the hardware implementation overcomes the SW version significantly. The bigger the population, the higher the acceleration, while the most advantageous choice of the population size is a multiple of the VRC count.

The coevolutionary design performance of the hardware platform was compared to a software implementation, again

Table 3: Hardware platform evolutionary process performance (10,000 generations, image size  $256 \times 256$  pixels, 1-5 mutations per chromosome) obtained by running 100 independent tests for each population size.

population size	5	10	15	20	25
SW time (s)	30.83	74.65	122.39	183.40	233.71
HW time (s)	7.21	7.86	14.17	14.44	14.83
acceleration	4.28	9.50	8.63	12.70	15.77

optimized using OpenMP and SSE 4.1 instructions. Because of two evolutionary processes running in parallel and very poor data locality, the performance of the software implementation was vastly degraded. Therefore the speed-up

Table 4: Coevolutionary design performance.

FCS size	50 %	25 %	12.5 %
SW time (s)	713.23	405.47	223.18
HW time (s)	12.51	6.91	4.23
acceleration	56.99	58.64	52.82
FCS size	6.25 %	3.125 %	1.5625 %
SW time (s)	133.91	88.26	71.92
HW time (s)	3.57	4.20	3.97
acceleration	37.49	21.04	18.11

is much more significant in the case of coevolutionary design. Table 4 shows performance tests results for software and hardware approaches. The experimental setup was as follows: 10,000 generations, population of 20 individuals, image size  $256 \times 256$  pixels, 1-5 mutations per CGP chromosome, up to 2 % mutations per FCS. Note that for FCS sizes lower than 12.5 %, the evolution time is similar. Due to very low fitness cases count, the fitness computation time  $t_f$  is shorter than the mutations time  $t_m$  and hardware utilization goes down. Moreover, the FCS evolution runs faster due to lower overhead and the FCS is updated more often. That is why the computation time can surprisingly grow with FCS size decrease.

## Conclusions

In this paper, a hardware platform for coevolutionary CGP speed-up based on FPGA technology has been proposed. Two-population coevolutionary algorithm running on dual MicroBlaze soft processor system has been accelerated using custom peripheral based on virtual reconfigurable circuit approach. The full pipelined VRC along with a special fitness cases memory enables very efficient fitness calculation. The performance of the hardware was experimentally evaluated in the task of evolutionary image filter design. It was shown that using custom hardware, universal processor throughput can be greatly overcome in the task of the evolutionary design and even more in the coevolutionary case. Various sizes of fitness cases subset have been applied to demonstrate the coevolutionary approach benefits. Especially for higher noise intensities, reduction of the FCS size leads to better results.

With small modifications, the hardware platform can be used to effectively evolve other digital circuits using coevolutionary CGP. In our future work, we will focus on designing image filters for other noise types as well as other image transformations, combinational logic design and other tasks suitable for coevolutionary design.

## Acknowledgements

This work was supported by the Czech science foundation project P103/10/1517 and the BUT project FIT-S-11-1.

## References

- Dobai, R. and Sekanina, L. (2013). Towards evolvable systems based on the Xilinx Zynq platform. In *2013 IEEE International Conference on Evolvable Systems (ICES)*, pages 89–95. IEEE Computational Intelligence Society.
- Gonzalez, R. C., Woods, R. E., and Eddins, S. L. (2009). “Standard” test images. *ImageProcessingPlace.com*. [online]. <http://www.imageprocessingplace.com/>.
- Hillis, W. D. (1990). Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D*, 42(1):228–234.
- Imamura, K., Foster, J. A., and Krings, A. W. (2000). The Test Vector Problem and Limitations to Evolving Digital Circuits. In *Proc. of the Second NASA/DoD Workshop on Evolvable Hardware*, pages 75–79. IEEE Computer Society.
- Miller, J. F., editor (2011). *Cartesian Genetic Programming*. Natural Computing Series. Springer Verlag.
- Popovici, E., Bucci, A., Wiegand, R., and De Jong, E. (2012). Co-evolutionary principles. In *Handbook of Natural Computing*, pages 987–1033. Springer Berlin Heidelberg.
- Schmidt, M. D. and Lipson, H. (2008). Coevolution of Fitness Predictors. *IEEE Transactions on Evolutionary Computation*, 12(6):736–749.
- Sekanina, L. (2003). *Evolvable Components - From Theory to Hardware Implementations*. Natural Computing Series. Springer Verlag.
- Sekanina, L., Harding, S. L., Banzhaf, W., and Kowaliw, T. (2011). Image processing and CGP. In *Cartesian Genetic Programming*, pages 181–215. Springer Verlag.
- Sikulova, M. and Sekanina, L. (2012a). Acceleration of evolutionary image filter design using coevolution in cartesian GP. In *Parallel Problem Solving from Nature - PPSN XII*, LNCS 7491, pages 163–172. Springer Verlag.
- Sikulova, M. and Sekanina, L. (2012b). Coevolution in cartesian genetic programming. In *Genetic Programming*, LNCS 7244, pages 182–193. Springer Verlag.
- Vanneschi, L. and Poli, R. (2012). Genetic programming – introduction, applications, theory and open issues. In *Handbook of Natural Computing*, pages 709–739. Springer Berlin Heidelberg.
- Vasicek, Z. and Sekanina, L. (2010). Hardware accelerator of cartesian genetic programming with multiple fitness units. *Computing and Informatics*, 29(6):1359–1371.
- Vasicek, Z. and Sekanina, L. (2011). Formal verification of candidate solutions for post-synthesis evolutionary optimization in evolvable hardware. *Genetic Programming and Evolvable Machines*, 12(3):305–327.



# Coevolutionary Dynamics Caused by Asymmetries in Predator-Prey and Morphology-Behavior Relationships

Takashi Ito<sup>1,2</sup>, Marcin L. Pilat<sup>1</sup>, Reiji Suzuki<sup>1</sup> and Takaya Arita<sup>1</sup>

<sup>1</sup>Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan

<sup>2</sup>takashi@alife.cs.is.nagoya-u.ac.jp

## Abstract

Predator-prey interactions are the key element of ecological systems. We present the results of morphology-behavior predator-prey coevolution in a 3D physically simulated environment. The morphologies and behaviors of virtual creature predators and prey are evolved using a genetic algorithm and random one-on-one encounters in a shared environment. There are two levels of asymmetries in the model: One is between two species, predators and prey, and the other is between two traits in each species, morphology and behavior. We analyze and discuss the complex coevolutionary dynamics caused by the asymmetries on the basis of quantitative characterization of morphology and behavior.

## Introduction

Predator-prey interactions are the key element of ecological systems (Legreneur et al., 2012). Predation pressures in food chains shape diversity and functions of organisms (Agrawal, 2001). Many predators employ various strategies capturing their prey, and at the same time, many prey employ various protective mechanisms against their predators (Edmunds, 1974). These strategies arose through the coevolution between predators and prey. Furthermore, in the coevolution, morphology and behavior have been tightly coupled in each species. Therefore, the process can be regarded as double coevolution of morphology-behavior and predator-prey couplings (Fig. 1).

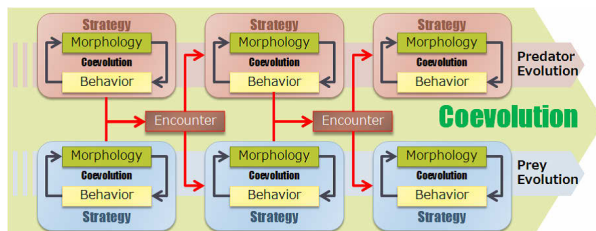


Figure 1: Double coevolution of morphology-behavior and predator-prey.

Predator-prey systems are conventionally studied in mathematical biology (population dynamics) using mathematical

methods (e.g., Lotka-Volterra equations) by analyzing evolutionary changes at the population level (Murray, 2003). However, these studies have not focused or have not been able to focus attention on coevolution of morphology and behavior of individual virtual creatures. On the other hand, virtual creature models in Artificial Life, following the pioneering study (Sims, 1994b), allow us to analyze the morphology and behavior coevolution in 2D and 3D environments (Ventrella, 1998; Taylor and Massey, 2000; Chaumont et al., 2007; Miconi and Channon, 2006; Pilat and Jacob, 2008; Turk, 2010; Azarbadegan et al., 2011). Additionally, due to the physical nature of the simulation, we are able to compare the resulting virtual creatures with biological organisms sharing similar morphological and behavioral traits (Chaumont et al., 2007). Some studies explored competitive coevolution in this framework (Sims, 1994a; Miconi, 2008). However, they have not focused on the morphology-behavior coevolution under the predator-prey coevolution and have not analyzed the strategies based on their morphological and behavioral characteristics.

The purpose of our study is to understand the evolutionary dynamics of the predator and prey strategies emerging in the context of this double coevolution. We perform double coevolution of morphology-behavior and predator-prey couplings by using a simple predator-prey scenario in a 3D physically simulated environment. As a first step of our investigation, we observed the emergence of various morphological and behavioral prey defensive strategies and found a weak tendency for the order of strategy emergence between morphologies and behaviors by using cross-correlation methods in the simulated environment (Ito et al., 2012, 2013). In this paper we give a simple quantitative characterization of morphology and behavior of each virtual creature. We track the evolutionary changes in special indices for both predators and prey species, specifically through the evolutionary process in which prey evolve one of the defense strategies, “Guard Strategies”. We then discuss the coevolutionary dynamics in terms of the asymmetries in the predator-prey and morphology-behavior relationships.

## Model

We use the Morphid Academy open-source simulation system (Pilat and Jacob, 2008) to evolve virtual creatures in a 3D physically simulated environment (Fig. 2). Morphid Academy has previously been used to successfully evolve virtual creatures for locomotion (Pilat and Jacob, 2008), light-following (Pilat and Jacob, 2010), and sustained resource foraging (Pilat et al., 2012). The presented coevolution of predators and prey provides an example of simulating several agents in a shared environment of Morphid Academy in a coevolutionary context.

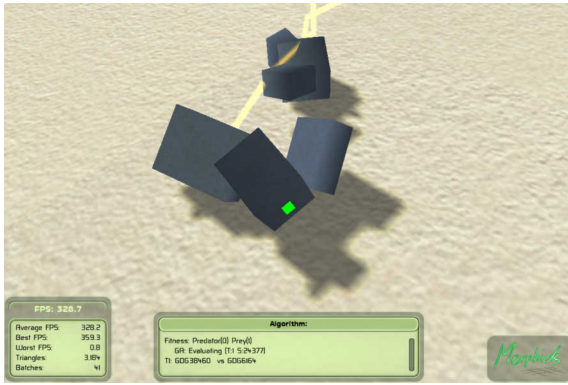


Figure 2: Virtual creatures evolved in Morphid Academy.

## Agents

The agents are virtual creatures comprised of several 3D rectangular solid body parts connected with simple hinge joints. Their physical phenotype is developed from a directed graph (Fig. 3). The nodes represent body parts and the links represent joints.

The genotype graph undergoes evolution through a genetic algorithm. We termed the root body part as the *torso*, and all the other parts as *limbs*.

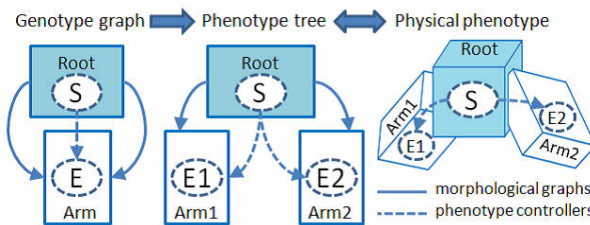


Figure 3: The development from genotype to phenotype.

The controller of a virtual creature is a recurrent neural network embedded in body nodes. There are three types of neurons: input, calculation and output. The input neurons represent sensory information from the environment,

the computational neurons process the input and the results are fed into the output neurons as joint effectors that power the joints, making the creature move. The creature sensor detects other living agents nearest to the virtual creature within a sensing range  $r$ . This virtual creature model is a simplification of Sims' Blockies model (Sims, 1994b) and is fully described in (Pilat and Jacob, 2008). The simplification in body and neural structure decreases the evolutionary search space and has been demonstrated to perform well for various evolutionary tasks.

## Experimental Environment

To represent a predator-prey encounter, we simulate a single prey creature with a single predator creature in a shared environment. A random prey creature is positioned near the origin of the simulation space. A random predator is then randomly positioned at  $r_0$  distance from the prey as shown in Fig. 4. Both agents are positioned above the simulation plane and allowed to free-fall due to gravity during a stabilization phase. Once they are stable from movement and resting on the ground surface, the evaluation encounter begins and lasts for  $S$  simulation time steps. Capturing is defined as the predator touching the torso of the prey with any of the predator's body parts. A captured creature is disabled and cannot be sensed.

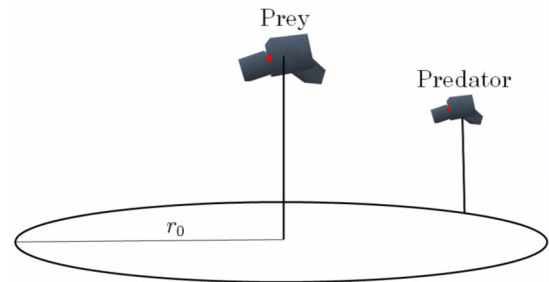


Figure 4: Initial positions of a predator and a prey at the start of the encounter.

## Evolution

Two populations are concurrently evolved, representing the predators and the prey. A steady-state genetic algorithm is used with tournament selection of 3 predator-prey pairs.. Fitness of each agent is calculated from the result of an encounter between the randomly selected predator-prey pair. For each tournament, one or two individuals, per population, with the best fitness can produce a child through one of the genetic operators of copy, crossover or grafting. The child replaces the worst performing of the 3 individuals of the corresponding population (prey or predator). Mutation is applied to the resulting child individual and includes: mutation of the morphological nodes or link parameters, addition

of morphological nodes, and the addition or removal of morphological links.

### Fitness Functions

The fitness of each agent is calculated after the predator-prey encounter by a fitness function. The fitness of a predator is defined by Eq. 1. Fitness of 5000 is allocated if the predator has captured the prey with an additional maximum of 5000 points proportional to the capture time. If the prey is not caught, the fitness is proportional to the distance gained towards the prey, based on the initial distance  $r_0$  and the final distance  $r_n$ .

$$F_{pred} = \begin{cases} 5000 + 5000 \times \frac{t-t_n}{t} & (caught) \\ 5000 \times \frac{r_0-r_n}{r_0} & (missed, r_0 \geq r_n) \\ 0 & (missed, r_0 < r_n) \end{cases} \quad (1)$$

The fitness of the prey is defined by Eq. 2. If the prey escaped from the predator (not caught within the specified simulation time), it receives a fitness value of 5000 with an additional value of up to 5000 points proportional to the distance it moved  $l_n$ . Otherwise, the fitness is calculated according to the ratio of the time the prey escaped during  $t_n$  over the time limit  $t$ .

$$F_{prey} = \begin{cases} 5000 + 5000 \times \frac{l}{l_n} & (escaped, l_n \leq l) \\ 10000 & (escaped, 0 \leq l < l_n) \\ 5000 \times \frac{t-t_n}{t} & (caught) \end{cases} \quad (2)$$

### Morphological and Behavioral Indices

Two simple indices are used to characterize morphologies and behaviors of virtual creatures quantitatively and to track their evolution. We use the ratio of the volume of the torso to the total volume as a morphology index (hereafter referred to as MOR). The reason for this is that Guard Strategies are characterized to have big or many limbs protecting a small torso from being captured (Ito et al., 2012). It is more difficult to quantitatively characterize the behavior of virtual creatures to represent the progress of evolution since, in general, the behavior heavily depends on the morphology, which itself is difficult to quantitatively characterize. After conducting preliminary experiments using many candidate indices, we decided on a simple index: the average output of effector neurons (hereafter referred to as BEH), which is intended to approximately represent the mobility of virtual creatures approximately, and does not depend directly on the agent morphology.

## Result

### Parameters

We evolved predator and prey populations, each of size  $i = 30$  and initially random individuals, for  $g = 10000$  tournaments. Each evaluation of an encounter was performed for  $S = 100000$  simulation time steps with an initial distance  $r_0 = 700$  between the agents. For each tournament,

a child was created by asexual copy (probability of 40%), crossover (30%), or grafting (30%). Mutation of the child was performed with prob. of 80% with each mutation able to apply small changes to the whole genome (prob. of 5% per gene). The vision radius of predators was 5000 while the prey were only able to see within 500 distance units. Therefore, the predator is able to sense the prey much earlier than the prey.

In previous studies, we classified the evolved prey's defensive strategies into two types, each with an assortment of evolved morphologies and behaviors: Runaway Strategy which involves fleeing from the predator and Guard Strategies (*Turtle*, *Clam* and *Tower* types) which rely on their morphologies and typically stationary behaviors to provide protection from predation (Ito et al., 2012). It is easy to detect the emergence of the Guard Strategies as they tend to evolve with a sharp increase in the fitness. Therefore, we investigate the relationship between morphology and behavior evolution by focusing on the course of the evolution of Guard Strategies. To control the movement of prey and to promote the emergence of Guard Strategies, we used the modified fitness function of the prey (Eq. 2) and the environmental parameter  $l = 100$ .

### Coevolutionary Dynamics

We performed 30 trials, among which we observed that prey evolved some Guard Strategy (Fig. 5) to prevent predator capture in 17 trials. The prey did not seem to have evolved any defensive strategies in the other trials. We further observed 12 trials with a clear increase in fitness out of the 17 trials. Each of the 12 trials evolved a specific prey strategy: Tower (6), Clam (4), and Turtle (2).

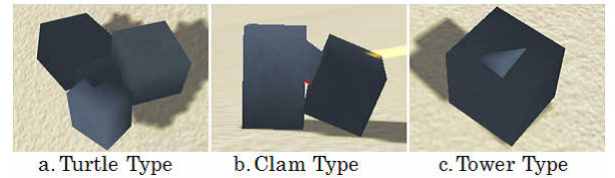


Figure 5: Sample morphologies of the Guard Strategies.

Fig. 6 shows a typical evolution trial in which *Clam* type Guard Strategy emerged. The blue and red lines in the middle graph represent the average fitness of the prey and predators, respectively. Each of the top and bottom graphs (top: prey, bottom: predators) represents a distribution of virtual creatures in the space defined by the two indices (X-axis: MOR, Y-axis: BEH, both in a logarithmic scale after normalization). In these graphs, a circle represents an individual, with its radius proportional to its fitness. We can estimate that the individuals close to each other have similar phenotypes. Specifically, two individuals sharing the same X-coordinate have the similar morphological characteristics,

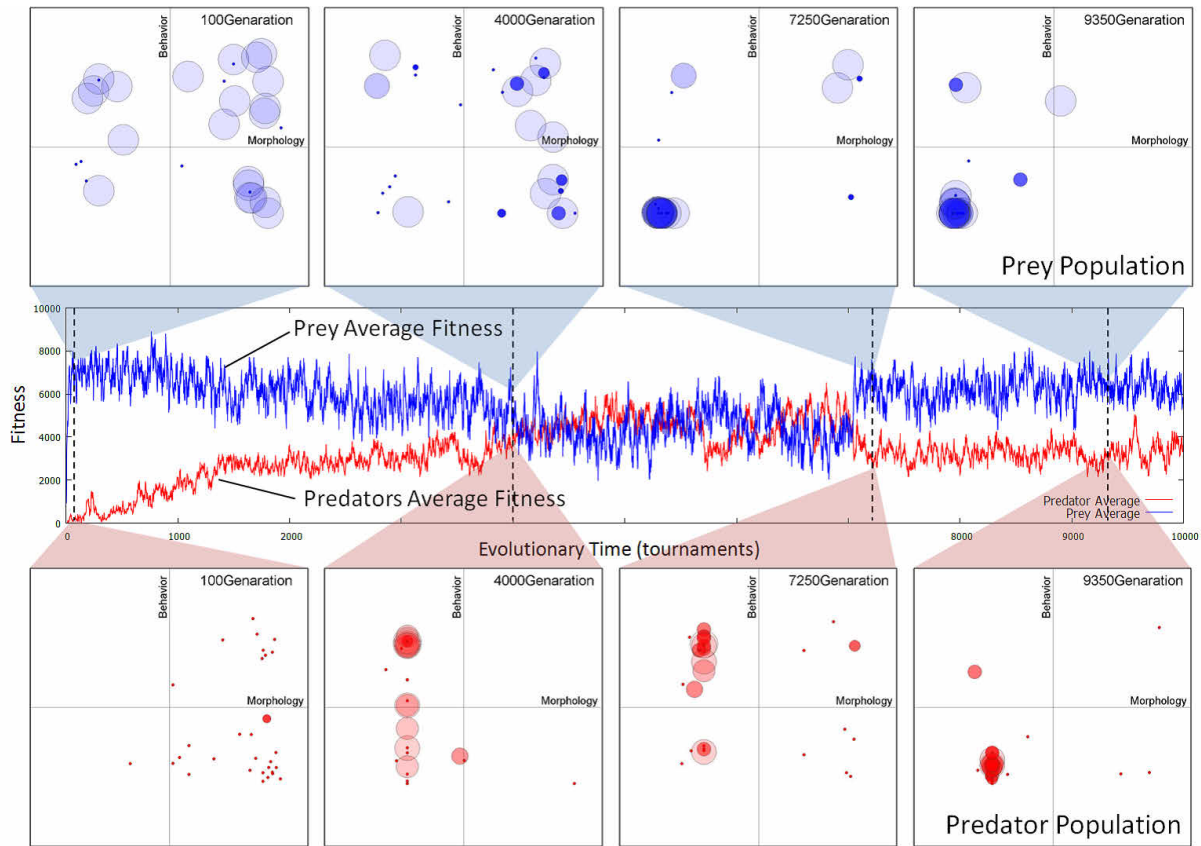


Figure 6: The evolution of the average fitness and the distributions of individuals on the trait space.

and sharing the same Y-coordinate corresponds to the similarity of their behaviors.

In early generations, the average fitness of the predator (prey) population was very low (high), which is also shown by the small (large) radius of the circles on the corresponding trait space. It corresponds to the situation in which the predators could not catch the prey at all. A wide distribution of the circles on the trait space of predators or prey reflects a wide variety of morphologies and behaviors of randomly generated individuals.

After that, until around the 4500<sup>th</sup> generation, predator fitness gradually increased while prey fitness slightly decreased. Large circles on the predator trait space, which represent the evolved predators, can be seen to create an elongated cluster, with data points having similar X-coordinates and a large variety of Y-coordinates. This indicates that the effective strategies that emerged in this predator population have similar morphological characteristics but diverse behavioral characteristics. In the trait space of the prey, the number of individuals with high fitness decreased while individuals with low fitness increased, as a consequence of the predator population's evolution. The tendency for some in-

dividuals of the prey population to maintain high fitness in this generation may be simply due to them encountering incompetent predators.

At some point near the 7000<sup>th</sup> generation, a switch in fitness performance occurred. This was due to the emergence of individuals with a strong defensive Guard Strategy, as shown in Fig. 8, which suddenly took over the population, represented as the reduction of the circle distribution on the trait space. Notice that when comparing the distribution of predators of the 7250<sup>th</sup> generation with that of the 4000<sup>th</sup> generation, there are many circles whose size decreased while their positions were the same. This means that although the major strategy of the predators (depending on a specific morphology) did not change between these two generations, they failed to catch the prey adopting the emerging Guard Strategy.

Finally, the cluster of the predators on the trait space shifted downward in 9350<sup>th</sup> generation. It means that the predators changed their strategy by adjusting their behaviors to adapt to the prey strategy. However, the prey population kept high fitness with their Guard Strategy unchanged.



### What Happened when a New Strategy Emerged?

Here we present a method to investigate the dynamics of morphology-behavior coevolution. We track the evolving populations especially when Guard Strategy emerged, by sequentially plotting the average position of the population of predators or prey on the trait space. The fitness-weighted center is used when calculating an average position. Note that we ascribed the fitness to the weight when we calculated the center of the mass. What we see from the coevolutionary trajectory and the fitness transitions includes the evolutionary order of the emergences of new or improved morphology and behavior.

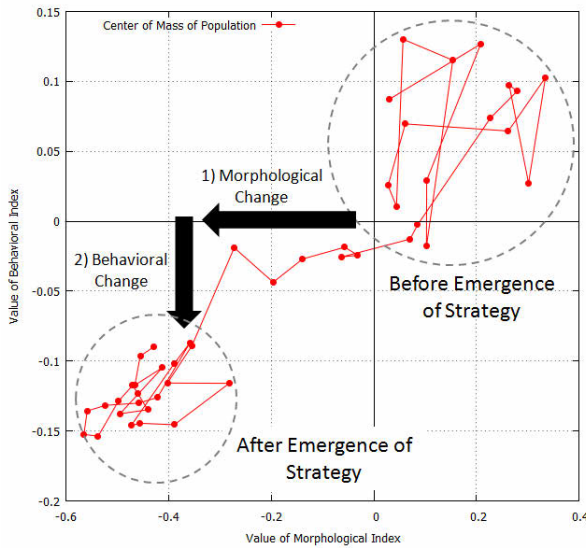


Figure 7: A typical evolutionary trajectory on the trait space when Guard Strategies (*Clam* and *Turtle* types) evolved in the prey population.

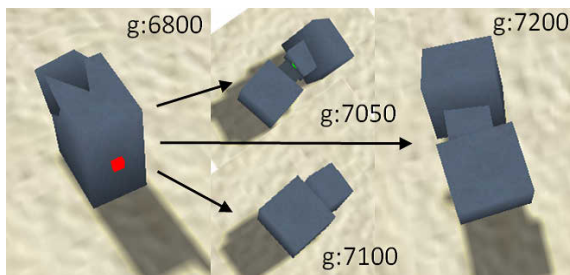


Figure 8: The virtual creatures with the best fitness in each generation appearing in the evolution trial shown in Fig. 7. The right three individuals have obtained the guard strategy and are the offspring of the left one.

Fig. 7 and 9 show the two typical evolutionary trajectories

when Guard Strategies emerged. The horizontal and vertical movements correspond to the morphological and behavioral changes of the prey population, respectively. Therefore, we can estimate that Fig. 7 shows that the morphological changes preceded the behavioral changes when the defense strategy was acquired, in other words, the morphological characteristics of the strategy spread in the population before behavioral characteristics spread. Fig. 8 shows an ancestor and its three offspring prey with Guard Strategy in this evolution trial, all of which obtained the best fitness in each generation.

On the contrary, Fig. 9 shows roughly that the behavioral changes preceded the morphological changes. Fig. 10 shows an ancestor and its three offspring prey with Guard Strategy in this evolution trial, all of which obtained the best fitness in each generation. They are characterized as its immovability as compared with the creatures shown in Fig. 7.

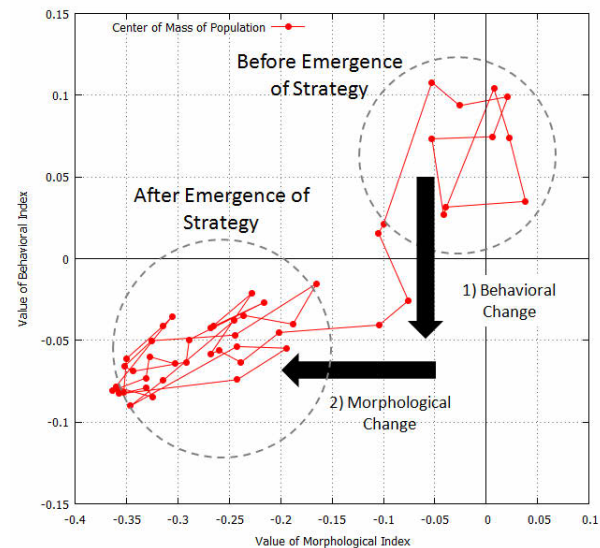


Figure 9: A typical evolutionary trajectory on the trait space when Guard Strategies (*Tower* type) evolved in the prey population.

It should be noticed that, as a general tendency of the step-wise evolutionary process, the fitness did not increase necessarily in accordance with the evolution of morphology or behavior. Instead, it clearly increased when either morphology or behavior was improved, which followed the evolution of the other. This observation tells us about close coupling between morphological and behavioral evolutions in the emergence of new strategies.

The difference in the evolutionary order of new trait emergence at least partly depends on the characteristics of evolved strategies. Guard strategies can be classified into the mobile and immobile ones. The former and the latter

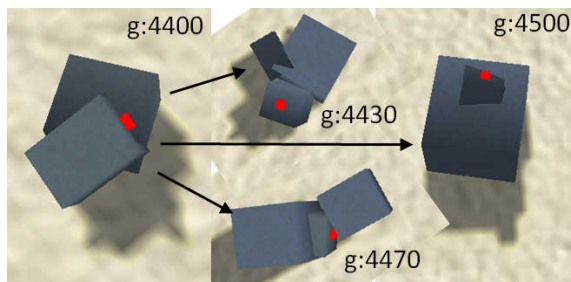


Figure 10: The virtual creatures with the best fitness in each generation appearing in the evolution trial shown in Fig. 9. The right three individuals have obtained the guard strategy and are the offspring of the left one.

correspond to the creatures in Fig. 8 (*Clam* and *Turtle* types) and Fig. 10 (*Tower* type), respectively. In general, immobility tends to need a specific sort of behavior while keeping some degree of freedom of their morphology. This might be the cause of the particularity of the emergence of the *Tower* type strategy in the context of the evolutionary order of the trait evolution. We will discuss the relationships between asymmetric properties and evolutionary dynamics in more general sense in the next section.

## Discussion

This paper demonstrates that we can observe the complex coevolutionary dynamics that takes place concurrently at predator-prey and morphology-behavior levels in a 3D physically simulated world. We discuss here that its complexity arises at least partly from the asymmetric properties in these relationships.

### Dynamics Caused by Population-Level Asymmetry

The predators were required to perform the following sequence of behaviors in the experiments: detecting a prey, approaching it, and then touching its torso part. Especially, successful methods for approaching need better morphology and behavior realizing faster movement and quick direction change, despite the fact that the bodies of both predators and prey are constructed and controlled by the same rules described in the Model section. In contrast, a prey adopting a simple strategy, such as moving forward in some direction, could be rather strong. Therefore, in general the selection pressure is considered stronger in the predator population than in the prey population. This asymmetry is not specific to our model but universal in natural ecosystems.

In the experiments, we frequently observed a typical coevolutionary dynamics “arms race” in which two evolving populations reciprocally drive one another to increasing levels of complexity. It was clearly shown that the asymmetry in the difficulty of required strategies caused the preceding evolution of the predators. The coevolutionary dy-

namics always started with the acquisition of the capturing strategies by predators as described in the previous section. This is due to the fact that the selection pressure acted more strongly on the predators in the initial stage in which both populations were occupied with randomly generated strategies. The prey started devising strategies in response to the increase in the selection pressure caused by the emergence of the effective strategies of the predators. A typical evolutionary scenario following the initial stage was regarded as a step-by-step evolution composed of several repetitions of alternate strategy improvement corresponding to the previous improvement of the other population. We also found in the experiments that the required time for predators to find or improve their strategies tended to be longer than in the case of prey. This is also due to the asymmetry between the predator-prey relationships.

### Dynamics Caused by Individual-Level Asymmetry

When comparing the changes in morphology and those in behavior, our impression, based on the experience with the coevolutionary experiments, is that a morphological change (e.g. the loss/growth of the limbs) tends to bring about a drastic change in the strategy. If this hypothesis is correct, a change in morphology and that in behavior in evolution correspond to an operator for global search and that for local search in the context of optimization problems. In other words, the morphology evolution has a potential to be a driving force to break away from the stalemate and in contrast, the behavioral evolution has a role to adjust the performance of the current strategy.

This hypothesis agrees well with the experimental results as follows. As for prey strategy evolution, we usually observed an emergence of an effective defense strategy characterized by a unique morphology, accompanied with a sharp increase in fitness. As for predator strategy evolution, we observed an emergence of a novel morphology that was followed by a gradual behavior evolution with the diversity in behavior, producing a gentle increase in fitness. However, only in the case of responding to the emergence of a novel prey strategy, behavior evolution played a great role compared with morphology evolution. These results can be further generalized as a hypothesis for morphology-behavior coevolution. Morphology evolution tends to precede behavior evolution in the case the evolution is rather independent of the other species evolution, while behavior evolution tends to precede or work dominantly in the case the evolution is responsive to a novel strategy evolved in another species. This hypothesis is also supported by previous results (Ito et al., 2013).

## Conclusion

We presented the results of evolutionary experiments investigating morphological and behavioral dynamics of a coevolutionary predator-prey scenario in a 3D physically simu-

lated environment. We defined a number of indices to quantify aspects of the morphologies and behaviors of our creatures, and used these to analyze the coevolutionary dynamics. The evolutionary dynamics of the strategies showed an “arms race” between the predators and prey, a typical feature of natural coevolutionary scenarios. We also temporally analyzed the coevolutionary dynamics of the morphology and behavior, focusing on the order in which new traits emerge. Our results illustrate how double coevolution, between predator and prey on the one hand and morphology and behavior on the other, can lead to asymmetrical development of morphology and behavior at both the intra-species level and the inter-species level. These two asymmetries led to complex coevolutionary dynamics in our 3D physical simulated framework, and likely do so in predator-prey interaction scenarios in general, both in artificial frameworks and in nature.

Our model could be extended in various directions. One obvious direction would be to use a scenario with many-to-many encounters. Such evolutionary experiments may shed light on the origin of group hunting and prey herding behaviors that are prevalent in the biological world (Sumpter, 2011). Furthermore, this direction might add a new direction to understanding the effect of the phenotypic change on the population dynamics (Pimentel, 1961; Rosenzweig and MacArthur, 1963). Another direction would be to investigate how the dynamics of double coevolution concentrated on in this paper can be applied in the field of engineering, including evolutionary robotics.

## References

- Agrawal, A. (2001). Phenotypic plasticity in the interactions and evolution of species. *Science*, 294(5541), page 321.
- Azarbadegan, A., Broz, F., and Nehaniv, C. L. (2011). Evolving sims’s creatures for bipedal gait. In *Proceedings of 2011 IEEE Symposium on Artificial Life (IEEE ALIFE 2011)*, pages 218–224.
- Chaumont, N., Egli, R., and Adami, C. (2007). Evolving virtual creatures and catapults. *Artificial Life* 13, pages 139–157.
- Edmunds, M. (1974). *Defence in Animals: A survey of anti-predator defences*. Longman Group Limited.
- Ito, T., Pilat, M. L., Suzuki, R., and Arita, T. (2012). Emergence of defensive strategies based on predator-prey coevolution in 3d physical simulation. In *Proceedings of the 6th International Conference on Soft Computing and Intelligent Systems, and the 13th International Symposium on Advanced Intelligent Systems 2012 (SCIS-ISIS2012)*, pages 890–895.
- Ito, T., Pilat, M. L., Suzuki, R., and Arita, T. (2013). Alife approach for body-behavior predator-prey coevolution: Body first or behavior first? In *Proceedings of the 18th International Symposium on Artificial Life and Robotics (AROB18th)*, pages 551–554.
- Legreneur, P., Laurin, M., and Bels, V. (2012). Predator-prey interactions paradigm: a new tool for artificial intelligence. *Adaptive Behavior* 20(1), pages 3–9.
- Miconi, T. (2008). In silicon no one can hear you scream: evolving fighting creatures. In *Proceedings of the 11th European conference on Genetic programming (EuroGP’08)*, pages 25–36.
- Miconi, T. and Channon, A. (2006). Analysing co-evolution among artificial 3d creatures. *Artificial Evolution*, 3871, pages 167–178.
- Murray, J. (2003). *Mathematical Biology I: An Introduction*. Springer-Verlag.
- Pilat, M. L., Ito, T., Suzuki, R., and Arita, T. (2012). Evolution of virtual creature foraging in a physical environment. In *Proceedings of the 13th International Conference on the Simulation and Synthesis of Living Systems (ALIFE13)*. MIT Press.
- Pilat, M. L. and Jacob, C. (2008). Creature academy: A system for virtual creature evolution. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2008)*, pages 3289–3297.
- Pilat, M. L. and Jacob, C. (2010). Evolution of vision capabilities in embodied virtual creatures. In *Proceedings of the 12th annual Conference on Genetic and Evolutionary Computation Conference (GECCO 2010)*, pages 95–102.
- Pimentel, D. (1961). Animal population regulation by the genetic feed-back mechanism. *The American Naturalist*, 95, pages 65–79.
- Rosenzweig, M. L. and MacArthur, R. H. (1963). Graphical representation and stability conditions of predator-prey interactions. *The American Naturalist*, 97, pages 209–223.
- Sims, K. (1994a). Evolving 3d morphology and behavior by competition. In *Proceedings of the 4th International Works on Synthesis and Simulation of Living Systems (ALIFE IV)*, pages 28–39.
- Sims, K. (1994b). Evolving virtual creatures. In *21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 94)*, pages 15–22.
- Sumpter, J. T. D. (2011). *Collective Animal Behavior*. Princeton University Press.
- Taylor, T. and Massey, C. (2000). Recent developments in the evolution of morphologies and controllers for physically simulated creatures. *Artificial Life*, 7 (1), pages 77–87.
- Turk, G. (2010). Sticky feet: evolution in a multi-creature physical simulation. In *Proceedings of the 12th International Conference on the Simulation and Synthesis of Living Systems (ALIFE12)*, pages 496–503.
- Ventrella, J. (1998). Attractiveness vs. efficiency: How mate preference affects locomotion in the evolution of artificial swimming organisms. In *Proceedings of the Sixth International Conference on the Simulation and Synthesis of Living Systems (ALIFE6)*, pages 178–186.

# The Influence of Cell Type on Artificial Development

John Maher, Fearghal Morgan, Colm O’Riordan and Brian McGinley

National University of Ireland Galway, Ireland  
john.maher@nuigalway.ie

## Abstract

Two variants of biologically inspired cell model, namely eukaryotic (containing a nucleus) and prokaryotic (without a nucleus) are compared in this research. Experiments are designed to provide an understanding of how the evolved regulation of protein transport to and from the nucleus of the eukaryotic type cell gives rise to complex temporal dynamics that are not achievable in a prokaryotic cell.

A novel system of protein movement based on the process of nucleocytoplasmic transport observed in the biological eukaryotic cell is proposed. Nucleocytoplasmic transport is considered by biologists to be one of the most important factors when determining the developmental trajectory of a cell, as it allows for additional control of transcription factors entering the nucleus, thereby regulating gene activity.

Experiments contrast the ability of both cell models to generate protein patterns within the cytoplasm. Results demonstrate that the additional cell complexity of the eukaryotic does not impede the Gene Regulatory Networks control. For increasingly difficult tasks requiring precise temporal control the performance of the eukaryotic cell model outperforms the prokaryotic cell model. In addition, results demonstrate that the second level of regulation introduced by the transport process within the eukaryotic cell allows very precise control of gene activity and provides the EA with a source of heterochronic control not possible in prokaryotic-type cells.

## Introduction

Cells are the fundamental building blocks of all biological life. Two distinct groups of cell exist, eukaryotic and prokaryotic. Eukaryotic cells are distinguished by the separation of the cell into compartments, the most pronounced compartment, the nucleus contains the genome (Figure 1). The genome is decomposed into genes, which encode the blueprint for creating the organism. During the process of development gene expression results in the creation of proteins levels within the cells. Proteins within the cell direct and dictate the cell fate and define its final role within the organism.

In order to enable gene expression, specialised proteins termed *Transcription Factors* (TF) must bind the gene cis-regulation sites. In response to TF binding, the gene expres-

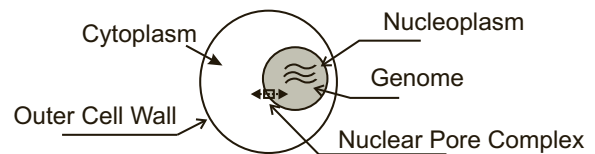


Figure 1: Architecture of the eukaryotic cell model. The nucleus is composed of the Nucleoplasm, Genome, and Nuclear Pore Complex (NPC)

sion rate is regulated, influencing protein levels within the cell. The presence of a nucleus within eukaryotic cells (Figure 1) restricts direct entry of transcription factors to the nucleus. Transport across into the nucleus through the *Nuclear Pore Complex* (NPC) is enabled by specialised *chaperons* proteins binding to TF proteins. The chaperon proteins are subdivided into two categories *importins* which enable import to the nucleus, and *exportins* which enable export from nucleus to cytoplasm across the NPC. *Importins* bind specific sites on the TF, named *Nuclear Localisation Sequence* (NLS). Export from the nucleus is enabled by the binding of *exportin* to *Nuclear Export Sequence* (NES) associated with the TF. This network of interacting genes and proteins is known as the Gene Regulatory Network (GRN).

Modifications to the relative timing of developmental events are termed *heterochronic*. Lee and Hannink (2003) report that the control of protein entry to the nucleus provides a powerful mechanism for the temporal regulation of gene expression. West-Eberhard (2003) highlights the importance of heterochronic control as a source of phenotypic novelty

“if I could control the time of gene action I could cause the fertilised snail egg to develop in an elephant”

Heterochronic change is not a developmental process but rather an evolutionary process (West-Eberhard, 2003). Modifications to the timing of events are heritable and must be somehow linked to the genetic encoding.

Artificial developmental systems have been introduced as a technique aimed at increasing the scalability of Evolutionary Algorithms (EA) (Haddow and Hoye, 2007). These



systems seek to solve the problem of scale by replacing the linear genotype-phenotype mapping, with a non-linear mapping. However, developmental mappings have a low degree of evolvability caused by high degrees of epistatic interactions at the genotypic level (Van Remortel et al., 2003). Enhancing the ability of the genome to allow for heterochronic mutations increases the number of successful genotypes (Stanley and Miikkulainen, 2003) by offering a variety of paths to each successful phenotype.

This paper presents an Artificial EvoDevo (AED) platform capable modelling evolutionary and developmental processes within biologically plausible eukaryotic and prokaryotic cells. The prokaryotic cell model is based on the work of (Kumar and Bentley, 2003). Experiments compare the ability of the evolved GRNs within the cell models to control gene expression for a number of static, periodic and aperiodic objectives. Results demonstrate an increased evolvability of eukaryotic cells compared to the prokaryotic cell model. Analysis of gene activity within the GRN shows that the transport process is instrumental in increasing evolvability by providing efficient heterochronic control of gene activity.

The structure of the paper is as follows: Section 2 provides a review of the use of developmental processes by the ALIFE community. Section 3 describes the Artificial EvoDevo (AED) platform developed, the eukaryotic cell model and its associated transport process. Section 4 describes a series of experiments and analyses the GRN dynamics of both eukaryotic and prokaryotic cell models. Section 5 concludes the paper.

## Background and Existing Research

This section provides a review of the use of developmental processes by the ALIFE community. Much of the research on developmental mappings has been motivated by the fact that the process of development is seen as a possible solution to the problem of scale in Evolutionary Algorithms (EA) (Bentley and Kumar, 1999), (Haddow and Hoye, 2007). By combining EAs with a developmental mapping between genotype and phenotype, the linear relationship between both is removed. Introducing developmental mappings also reduces the causality between genotype and phenotype spaces, which can reduce evolvability since regions of the search space become unreachable (Roggen and Federici, 2004).

The process of development is primarily a temporal one, where development starts from a single point and over time expands into a series of parallel pathways (Raff, 1996). Temporally shifting these processes relative to each other can give rise to phenotypic novelty. These shifts in the timing of events are termed *heterochronic mutations*, and must be heritable between generations. Efforts by the ALIFE community to identify sources of heterochrony within developmental encoding are limited. Matos et al. (2009) adapts

the framework proposed by Albrecht et al. to quantify the degree of heterochrony achievable by both grammar based and cellular ontogenies. At the biologically plausible GRN level, Banzhaf and Miller (2004) illustrate a possible genetic mechanism by which heterochrony can be enacted. This is due to an encoding of time and strength of gene expression into the strength of interaction between transcription factors and the genes cis sites. Kumar and Bentley (2003) suggest that the modification to the diffusion rate of signalling proteins can also provide a degree of heterochrony.

The developmental model proposed by Kumar and Bentley (2003) is the primary source of inspiration for the AED platform described in this paper. Using a prokaryotic cell model (Kumar and Bentley, 2003) demonstrate how intricate gene regulatory networks can be evolved to establish and control protein concentrations within a single cell. Furthermore they demonstrate the ability of the system to evolve 3D multicellular spherical morphologies.

This paper recreates the work of Kumar and extends it by introducing a eukaryotic type cell within the AED platform. Experiments contrast the evolved developmental processes within both cells by comparing the ability of the GRN to regulate protein concentrations for a variety of increasingly difficult static, dynamic and aperiodic objectives.

## The Artificial EvoDevo (AED) Platform

This section describes the developed Artificial EvoDevo (AED) platform, the eukaryotic cell model and its associated transport process. The section is decomposed into the following subsections - AED architecture and configuration, development and the cell cycle, protein model and classification, protein transport, mechanics of gene expression and the evolved genome structure.

The developmental algorithm within the AED platform captures the concepts of genes, proteins and cells. Similar to its biological counterpart, development proceeds along a time-line, where as a result of the GRN activity protein levels are established within the cell.

## AED Architecture and Configuration

The AED comprises two main components (Figure 2) the Genetic Algorithm (GA) and developmental algorithm. Evolved genomes are supplied by the Genetic Algorithm (GA) to the developmental algorithm. The genome is then developed by placing it inside a user selected cell type (eukaryotic or prokaryotic) and returns a fitness to the GA.

The user configures the AED via a configuration file (Table 1). In order to start the developmental process, maternal proteins are placed inside the cell. The biological counterpart of this process is fertilisation of the embryo. Seeding involves placing a single TF inside the cytoplasm. In the reported experiments this has been arbitrarily chosen as protein 0 with a concentration of 0.5. Eukaryotic type cells

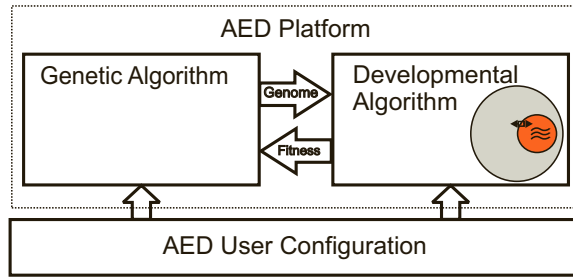


Figure 2: Architecture of the AED platform

are also seeded with *importins* (IM) and *exportins* (EX) type proteins at a concentration of 0.5.

### Development and The Cell Cycle

Having seeded the cell with maternal proteins evolved genomes are developed by executing a *cell cycle* Figure 3. The developmental process is executed for a user defined number of *steps* (Table 1a), by iterating the cell cycle shown in Figure 3. At the end of development, protein levels within the cell cytoplasm are used to determine the genotype fitness. The calculated fitness value is subsequently fed back to the GA (Figure 2), where the corresponding genome is then subject to evolutionary control.

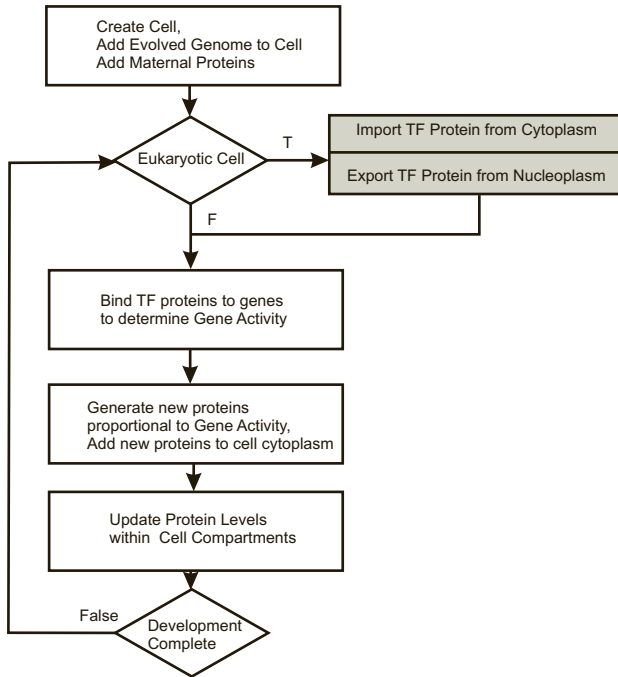


Figure 3: AED Cell Development Cycle

### Protein Model and Classification

The AED eukaryotic cell model contains three proteins classes (Table 2) while the prokaryotic model contains only

steps	200	Defines the # Development Steps
ntf	3	Number of TF's
nim	3	Number of Importins
nex	3	Number of Exportins

(a) Developmental Algorithm Configuration

runs	50	Number of GA runs
gens	4000	Number of generations per run
psize	100	Population size
pmut	0.1	Mutation Rate
tsize	20	Tournament Size
pxo	0.6	Probability of crossover
obj	Sin	Objective
tps	2	Number of test proteins

(b) Genetic Algorithm Configuration

Table 1: Typical Configuration Parameters used by the AED platform. The settings listed are typical for the experiments reported in this paper.

a single TF type protein. Details of the three protein classes are listed in Table 2. Proteins are distinguished by a *protein code* derived from the gene code (Figure 4). There is a direct mapping between gene code and protein code, ie. gene code 1 maps to protein code 1 etc. The relationship between protein IDs and protein class (TF, IM, EX) is determined by the user setting of *nim*, *nex* and *ntf* in the configuration file listed in Table 1a.

Protein Class	Description	ID Range
Transcription Factor (TF)	regulates gene activity	$0 \rightarrow ntf - 1$
Importin (IM)	enables TF import	$ntf \rightarrow ntf + nim - 1$
Exportin (EX)	enables TF export	$ntf + nim \rightarrow ntf + nim + nex - 1$

Table 2: Protein Class Names and their corresponding function. The ID range of the protein types are calculated based on the user configuration of Table 1

### Protein Transport

In eukaryotic cells TF type proteins must be first transported into the nucleus in order to regulate the rate of gene expression. TF proteins within the AED model include additional evolved NLS and NES regions, (Figure 4). Binding of IM/EX type proteins to these sites enables transport of the TF between compartments. During the transport phase of the cell cycle (Figure 3), each TF is selected and the proportion of protein exported  $C_{ex}(tf)$  and imported  $C_{im}(tf)$  is described by (1 and 2) respectively.

$$C_{ex}(tf) = TF_{nuc} * f(TF_{nuc} * W_{TF} + \sum_i^n (Ex_i * NES_i)) \quad (1)$$

$$C_{im}(tf) = TF_{cyt} * f(TF_{cyt} * W_{TF} + \sum_i^n (Im_i * NLS_i)) \quad (2)$$

where  $f$  is defined as the sigmoid function, with  $TF_{nuc}$  and  $TF_{cyt}$  being the concentrations of the selected TF in the nucleus and cytoplasm compartments respectively.  $W_{TF}$  is an evolved bias for the selected TF.

Export Control (NES)			4
0.1	-0.8	0.1	
0.2	-0.9	0.3	
Import Control (NLS)			Protein Code

Figure 4: The evolved NLS and NES sequences associated with each TF

### Mechanics of Gene Expression

Upon entering the nucleus the rate of gene expression (GE) is regulated by the binding of TF protein to the cis-regulatory sites of individual genes. For each gene encoded on the genome the expression rate  $GE_n$  is described as (3).

$$GE_n = SR_n * f\left(\sum_{i=0}^n I_n * TF_n - TH_n\right) \quad (3)$$

where  $SR_n$  is the evolved max synthesis rate for the gene,  $TH_n$  and  $I_n$  are the evolved gene threshold and interaction levels respectively. All protein produced during the gene expression phase is placed within the cytoplasm and any existing protein concentration ( $C_{n-1}$ ) is updated (4).

$$C_n = C_{n-1} - (C_{n-1} * DR) + GE_n \quad (4)$$

where  $DR$  is the evolved decay rate for this protein.

### The Evolved Genome Structure

The role of the GA is to provide candidate configurations (genomes) to the developmental algorithm. Following the development process these configurations are assigned a fitness. The GA is a derivative of the standard generational GA with elitism, gaussian mutation, uniform crossover and tournament selection. All parameters for the GA and development algorithms are user selectable via the configuration file (Table 1). The genome is subdivided into two chromosomes, with each chromosome subsequently decomposed into genes. The genes contained on the first chromosome exclusively encode the protein information for all 3 protein classes (TF, IM, EX). Genes on the second chromosome encode the transport specific information (NLS/NES) for each of the TF proteins (Figure 4). Figure 5 illustrates the structure of the genes contained on each chromosome.

### Experiments and Results

This section describes a series of experiments and analyses of the GRN dynamics of both the eukaryotic and prokaryotic models. The experiments compares the abilities of the GRNs within eukaryotic and prokaryotic models to

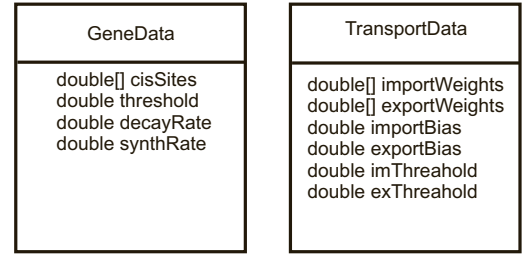


Figure 5: UML diagram of genes encoded on the 2 chromosome within eukaryotic cells.

evolve and follow defined protein patterns during their development. The objectives selected for the comparison are divided into three categories in order of increasing difficulty, namely static, periodic and aperiodic (Table 3). By contrasting the fitness achieved for the three classes of objective, allows a preliminary assessment of the contribution of transport within eukaryotic cell types.

Objective Name	Class
Lin	Static
Sin, Rect,	Periodic
Gauss, SinOff, GaussOff, AmpGauss	Aperiodic

Table 3: Objective Names and their corresponding classification

The simplest static objectives determine the fitness at the end of the developmental cycle, ignoring the concentration profile of the protein during the developmental cycle. For the static objective the fitness function is similar to that used by Kumar (5).

$$Fitness = \sum_{j=0}^{M-1} (C_j - (1 + j)/M)^2 \quad (5)$$

where  $M$  is the total number of transcription factor proteins under test and  $C_j$  is the concentration of protein  $j$  at the end of development.

In contrast both periodic and aperiodic objectives assess the fitness over the entire developmental time. Periodic objectives are designed to mimic their biological equivalent, termed circadian rhythms, where two proteins oscillate in lockstep. The aperiodic objectives, represent another biologically plausible objective as they place a precise temporal dependence on gene expression. In all test cases the number of proteins tested against the fitness function can be specified, up to the maximum number of transcription factors. Thus the fitness function for dynamic and aperiodic objectives is defined as (6).

$$Fitness = \sum_{j=0}^{M-1} \sum_{i=0}^n (O_{ji} - C_{ji})^2 \quad (6)$$

where  $C_{ji}$  is the concentration of protein  $j$  at time  $i$ , and  $O_{ji}$  is the objective concentration.

For each of the seven objectives tested the AED platform, was typically configured as per Table 1.

### Comparison of Eukaryotic and prokaryoticCell Dynamics

Results for each of the seven objectives (Table 3) are listed in Table 4. These results illustrate that both cell configurations can solve static and dynamic tasks with a high degree of accuracy. For the aperiodic objectives the eukaryotic type cells configurations show a considerable improvement over the prokaryotic cells. This improvement has its origins in the ability of the eukaryotic cell to limit the target protein activity to the specific times during the development. In contrast, the prokaryotic cells tends to have a continuous level of protein present at all times. Figure 7 illustrates the phenotype (protein levels within the cytoplasm) associated with each of the aperiodic objectives.

Fitness Results			
Objective	Nucleus	Best	Mean
Lin	T	0.0	0.092
	F	0.0	0.041
Sin	T	0.046	199.7
	F	0.058	250.95
SinOff	T	1.05	1288.71
	F	20.67	1415.8
RectSin	T	0.802	142.2
	F	1.31	131.13
Gauss	T	0.60	601.6
	F	4.765	785.39
GaussOff	T	0.855	590.25
	F	9.529	722.57
GaussAmp	T	0.581	391.78
	F	3.83	432.05

Table 4: Results for Objectives, the presence of a nucleus indicating a eukaryotic type cell is shown by the Boolean [T]rue.

### Results Analysis: Gene Activity

The section illustrates how the process of transport within the eukaryotic cell is instrumental in generating the protein profiles associated with the aperiodic genome solutions. The level of gene activity for each of the individuals developed in Figure 7 is plotted in Figure 8. Because the gene ID corresponds to the proteins ID any variation in gene activity results in a corresponding change in its protein level.

Gene activity for the prokaryotic cell configuration (Figure 8 b, d, f) shows continuous activity over the entire development time, which is penalised by the aperiodic objectives. The close coupling between the genes and proteins in the prokaryotic cell makes it difficult to generate the isolated the gene activity required for the aperiodic objectives. This

coupling arises as a consequence of the genome and proteins being contained in the same cell compartment.

In contrast the eukaryotic cell achieves very specific regions of gene activity for genes 0 and 1 with relative ease (Figure 8 a, c, e). The regions of activity for genes 0 and 1 are very localised to the required times for peak protein activity within the cytoplasm. Inspecting the gene activity for the IM genes, (IDs {3, 4, 5}) and EX genes (IDs {6, 7, 8}) shows very broad and intense levels of activity, indicating that the process of transport is very heavily involved in the generation of the target protein profile.

The dynamics of the *Gauss* profile in Figure 8 e deserve special mention, as there is little or no activity on Gene 0 around the peak presence of protein 0 in the cytoplasm. Figure 6 illustrates that the protein profile is generated as a result of exporting the stored protein 0 from the nucleus at the appropriate time, Figure 6.

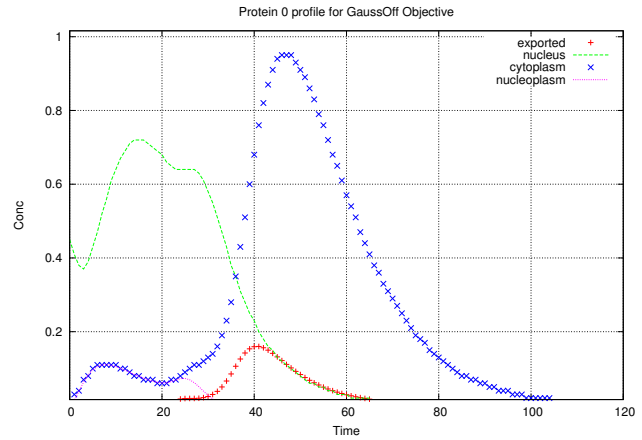


Figure 6: Protein 0 Dynamics for GaussOff Objective, illustrates the accumulation of Protein 0 in the nucleus prior to time step 20, while there is no gene activity around peak evaluation time, the cytoplasm protein profile for protein 0 is generated by exporting proteins from nucleus during this time.

### Results Analysis : Transport as a source of Heterochrony

This section investigates how mutations to the transport chromosome affect the quality of the aperiodic solutions. Selecting the best eukaryotic individuals from the *gaussOff* and *ampGauss* objectives illustrated in Figures 7 c, e the activity of the transport specific genes (ID 3-8) is reduced ('knocked out'), according (7) -

$$\text{GeneActivity} = KO * SR * \text{sigmoid}(\text{activity})$$

$$\text{where } KO \in A = \{0.9, 0.8, 0.5, 0.1\} \quad (7)$$

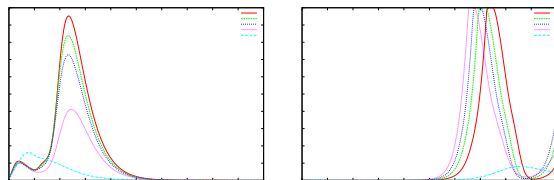
Figure 9 illustrates the full spectrum of heterochronic mutations are possible. For the *ampGauss* objective the level



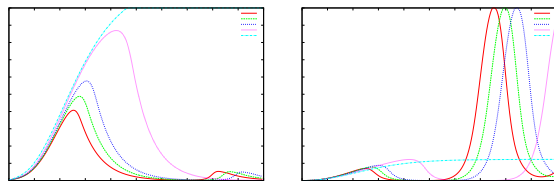
of protein 0 (Figure 9c), increases its level and duration in the cytoplasm in response knock out. The level of protein 1 remains relatively unaffected but its onset is delayed in response to knock out (Figure 9d).

For the *GaussOff* objective (Figure 9a) the level of protein 0 is reduced in response to increasing knock out while the onset of protein 1 occurs earlier in the development (Figure 9b).

In addition to reducing the activity level the transport specific genes, the NLS/NES interaction levels could also have targeted to give similar results. From an evolutionary perspective, mutations to the transport chromosome have a causal relationship to the generated phenotype allowing the conclusion that the addition of the transport process, which is in effect a second level of regulation tends to provide a smoothing of the phenotype landscape.



(a) Protein 0 GaussOff Objective, eukaryotic Cell Configuration (b) Protein 1 GaussOff Objective, eukaryotic Cell Configuration



(c) Protein 0 AmpGauss Objective, eukaryotic Cell Configuration (d) Protein 1 AmpGauss Objective, eukaryotic Cell Configuration

Figure 9: Heterochronic Mutations to the best eukaryotic cell individuals from the *GaussOff* and *AmpGauss* objectives, realised by reducing the activity of Importin and Exportin genes.

## Discussion and Conclusions

The regulated entry of *transcription factors* to the nucleus of eukaryotic type cells has been shown to have a major influence on the direction of biological development. This paper has reported a biologically inspired eukaryotic cell model that captures the concept of regulated protein transport to and from the cell nucleus. Tests on the evolvability of the GRN indicate that the addition of this level of complexity does not prevent the cell successfully generating GRN dynamics. Indeed, it serves to improve the GRNs ability to

evolve aperiodic objectives. Analysis of the gene activity within the eukaryotic cell shows that it relies heavily on the transport of TF to and from the nucleus to control gene activity. In particular it is observed that for aperiodic tasks TF protein is only present in the cytoplasm at the required development time intervals. In contrast, while the prokaryotic-cell model fared well for static and periodic tasks, its performance suffered significantly for aperiodic objectives. An examination of the gene activity within the prokaryotic-cell model has shown continuous levels of gene activity during development time. In contrast, the eukaryotic cell isolates its gene activity to very specific regions of the development time. The high levels of activity for the IM/EX genes indicates their importance in generating the protein dynamics. The eukaryotic cell model demonstrates the potential for heterochronic mutations to arise by scaling the activity of transport specific genes. Moreover a high degree of correlation between the level of disruption to these genes and the resulting change in protein profile has been observed.

## References

- Banzhaf, W. and Miller, J. (2004). The challenge of complexity. In *Frontiers of Evolutionary Computation*, pages 243–260. Springer.
- Bentley, P. and Kumar, S. (1999). Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In *Proceedings of the genetic and evolutionary computation conference*, volume 1, pages 35–43.
- Haddow, P. C. and Hoyer, J. (2007). Achieving a simple development model for 3d shapes: are chemicals necessary? In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1013–1020.
- Kumar, S. and Bentley, P. J. (2003). Biologically inspired evolutionary development. In *Evolvible Systems: From Biology to Hardware*, pages 57–68. Springer.
- Matos, A., Suzuki, R., and Arita, T. (2009). Heterochrony and artificial embryogeny: A method for analyzing artificial embryogenies based on developmental dynamics. *Artificial life*, 15(2):131–160.
- Raff, R. A. (1996). The shape of life: genes, development, and the evolution of animal form.
- Roggen, D. and Federici, D. (2004). Multi-cellular development: is there scalability and robustness to gain? In *Parallel Problem Solving from Nature-PPSN VIII*, pages 391–400.
- Stanley, K. O. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.
- Van Remortel, P., Ceuppens, J., Defaweux, A., Lenaerts, T., and Manderick, B. (2003). Developmental effects on tuneable fitness landscapes. In *Evolvible Systems: From Biology to Hardware*, pages 117–128. Springer.
- West-Eberhard, M. J. (2003). *Developmental plasticity and evolution*. Oxford University Press, USA.

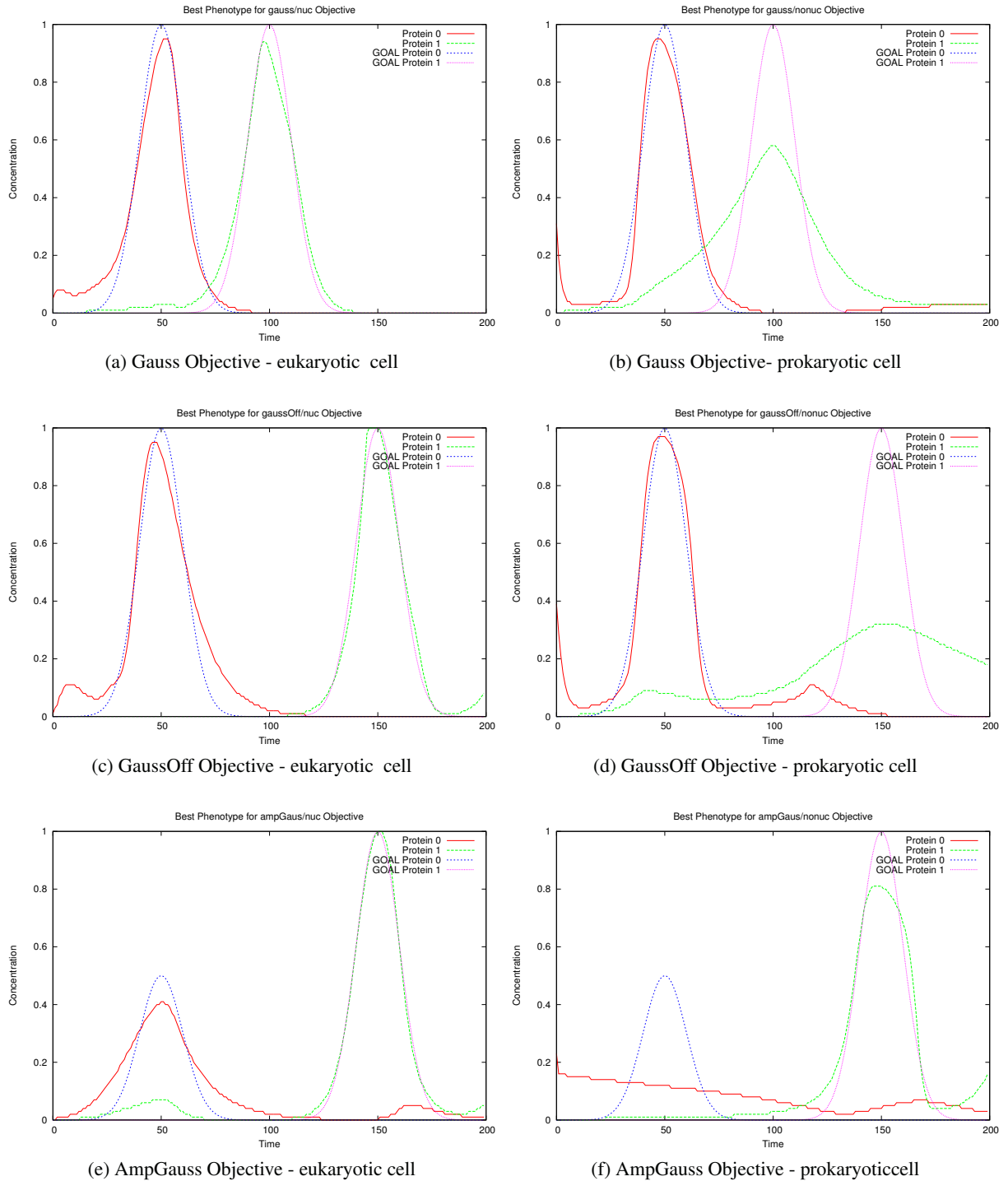


Figure 7: Protein profiles generated by the best Individuals as reported in 4 - Aperiodic objectives only, contrasted against the Objective. The AED configuration of Table 1 configures proteins IDs (0,1) to be used in the fitness calculation.

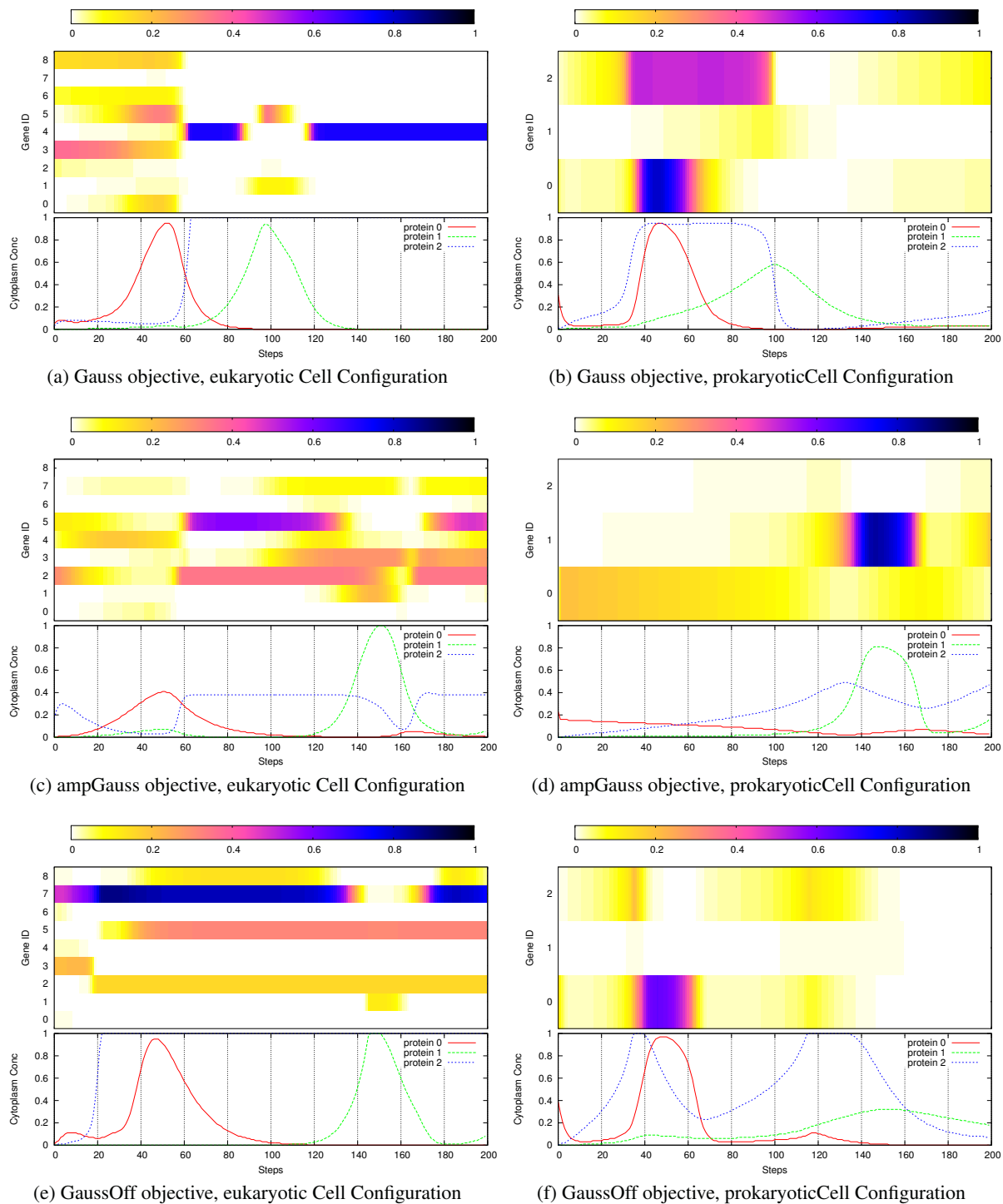


Figure 8: Gene Activity for best evolved individuals for aperiodic objectives. The top section of each plot maps the gene activity to a colour intensity, while the bottom section shows the corresponding protein profile in the cytoplasm. For subplots (a,c,e) the configuration listed in Table 1 configures the import proteins IDs range from 3-5, and export proteins IDs range from 6-8

# Controlling development and chemotaxis of soft-bodied multicellular animats with the same gene regulatory network

M. Joachimczak<sup>1</sup>, T. Kowaliw<sup>2</sup>, R. Doursat<sup>2,3</sup>, and B. Wróbel<sup>1,4</sup>

<sup>1</sup>Systems Modeling Laboratory, IO PAN, Sopot, Poland

<sup>2</sup>Institut des Systèmes Complexes Paris Île-de-France (ISC-PIF), CNRS, Paris, France

<sup>3</sup>School of Biomedical Engineering, Drexel University, Philadelphia, USA

<sup>4</sup>Evolutionary Systems Laboratory, Uniwersytet im. Adama Mickiewicza, Poznań, Poland  
mjoach@iopan.gda.pl

## Abstract

The ability to actively forage for resources is one of the defining properties of animals, and can be seen as a form of minimal cognition. In this paper we model soft-bodied robots, or “animats”, which are able to swim in a simulated two-dimensional fluid environment toward food particles emitting a diffusive chemical signal. Both the multicellular *development* and *behaviour* of the animats are controlled by a gene regulatory network (GRN), which is encoded in a linear genome. Coupled with the simulated physics, the activity of the GRN affects cell divisions and cell movements during development, as well as the expansion and contraction of filaments connecting the cells in the swimming adult body. The global motion that emerges from the dynamics of the animat relies on the spring-like filaments and drag forces created by the fluid. Our study shows that it is possible to evolve the animat’s genome (through mutations, duplications and deletions) to achieve directional motion in this environment. It also suggests that a “minimally cognitive” behaviour of this kind can emerge without a central control or nervous system.

## Introduction

In biological multicellular organisms, the dynamics of gene regulatory networks (GRNs) controls not only the growth of the organism, including the maintenance of the cells and overall structure, but also its behaviour. A striking example can be observed in social amoeba such as *Dictyostelium* (slime mold), where gene regulation controls both the aggregation of single cells into a slug and the adaptive behaviour of this collective entity (Bonner, 2008).

In theoretical biology and artificial life, artificial gene networks are used to understand how computational properties of biological networks evolve. One area of research is the evolution of control of multicellular development (Dellaert and Beer, 1996; Eggenberger Hotz, 1997; Doursat, 2009; Schramm and Sendhoff, 2011), another is computation in a more general sense (Banzhaf, 2003; Nicolau et al., 2010; Lopes and Costa, 2012). We addressed these two areas in our previous research using the artificial life system that we created, GReaNs (for Genetic Regulatory evolving artificial Networks; reviewed in Joachimczak and Wróbel, 2011). We investigated in particular the evolution of signal processing

using continuous or spiking computational units (Joachimczak and Wróbel, 2010; Wróbel et al., 2012), and the evolution of soft-bodied artificial organisms, or “animats”, whose development and locomotion were controlled by a GRN (Joachimczak and Wróbel, 2012; Joachimczak et al., 2012).

The use of a developmentally inspired stage to generate the morphology of a virtual robot is an active area of research, involving a range of abstractions for cellular and genetic control (Hornby and Pollack, 2002; Bongard and Pfeifer, 2003; Kowaliw et al., 2004; Doursat, 2008; Meng et al., 2011). The main contribution of our system lies in the combination of a biologically realistic encoding of the GRN (and genetic operators that allow for their complexification) with a realistic physics simulation. Physics rules govern the movement of cells during development, and the drag forces during locomotion in the fluid. Although in our current implementation the environment and the animats are two-dimensional, the system could be extended to 3D to make our results even more relevant. Physically plausible robots could take advantage of their softness—and thus resistance to damage and external forces—when interacting with other objects (for example, changing shape to squeeze through small openings). Although the properties of non-rigid, modular bodies have been explored before (Shimizu et al., 2005; Umedachi et al., 2010; Schramm and Sendhoff, 2011; Doursat et al., 2012; Hiller and Lipson, 2012; Rieffel et al., 2013), including our previous work on the diversity of locomotion strategies in soft-bodied animats (Joachimczak et al., 2012), the present study is the first attempt, to our knowledge, at evolving a fully decentralized controller and morphology of elastic animats that can sense and navigate their environment.

In the present paper we consider the evolution of gene regulatory networks able to control both the development of a soft-bodied animat and its emergent *multicellular chemotaxis*, a basic behaviour that consists of moving toward the source of an external signal. Despite its apparent simplicity, this task requires generating motion and coordinating numerous local cell actions to turn the body in the direction of a gradient. We identify and analyze here several morpholo-



gies and behavioural strategies toward this goal. The main contribution of this paper is to demonstrate that such *minimal cognition* (van Duijn et al., 2006) can collectively evolve in a multiagent system. We also suggest that it could be the first step toward more advanced cognitive abilities (Wróbel, 2012). Another novelty is a simplification of the artificial physics: instead of keeping a different set of environmental conditions for the developmental phase and the behavioural phase, we adopt the same physical rules for both.

## Controlling the development and behaviour of multicellular soft-bodied animats

The model used in this paper builds upon our previous work on soft-bodied multicellular animats (Joachimczak et al., 2012; Joachimczak and Wróbel, 2012). As before, the gene regulatory networks that control the bodies are encoded in linear genomes. We provide here only a brief summary of how the encoding works and how the dynamics of the network is simulated, then we describe in more detail two modifications that we brought to the system: unifying the physical conditions of the developmental and behavioral phases, and designing new cell-to-cell communication. In this new design, chemical signals diffuse between cells under a constraint of conservation of their total amount (in the previous implementation, mass was not conserved).

### Genome and gene regulatory network

The genome is represented by a list of genetic “elements” without fixed length. Genetic elements belong to three classes: (i) *genes*, which code for *products* (transcription factors or chemicals diffusing between cells), (ii) *regulatory elements*, and (iii) *special elements*, which encode inputs and outputs of the regulatory network. One or more regulatory elements form a regulatory region, which can be followed in the genome by one or more genes to make a *regulatory unit*. The activation levels of the regulatory elements of a unit influences the concentrations of products coded by this unit’s genes (Fig. 1). Conversely, regulatory elements are activated by the products currently present in the cell, which virtually “bind” to the genome with various probabilities (related to their concentration) and various affinities to the regulatory sites.

To simulate the behaviour of a cell, we first decode the genome to obtain the corresponding GRN, in which nodes represent regulatory units, and weighted directed edges represent relations of regulation. The signs of the weights indicate whether the regulation is excitatory or inhibitory, while the weights tune the chemical affinity between products and regulatory elements. The affinity also depends on the “distance” between two elements, calculated by construing each element as a point in an abstract 2D space of chemical interactions (not to be confused with the physical 2D space of the animat). The affinity is set to 0 if the distance is above a

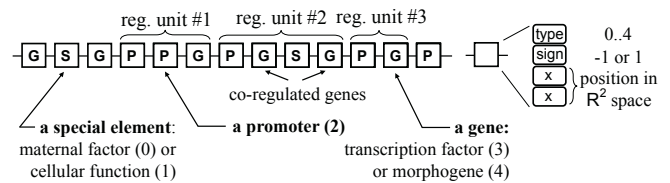


Figure 1: Genome (left) and structure of a genetic element (right). Each element consists of a *type* field, which specifies its class (G: gene, P: regulatory, S: special), a *sign* field, and  $N$  abstract *coordinates* (here,  $N = 2$ ), which determine its affinity to other elements based on distance in  $\mathbb{R}^N$ .

certain threshold, and to a maximum value if the two points overlap.

While each cell in the animat body contains the same GRN, the product concentrations that encode the dynamic state of this network can be different from cell to cell. Concentrations are real values updated in discrete time steps. The increase in concentration, or “synthesis rate”, of a product  $P$  is influenced by the concentrations of products that have a non-zero affinity to the regulatory elements of the unit encoding  $P$ . The combined effect of all the products binding to the same regulatory element is additive; the combined effect of all the regulatory elements in a regulatory unit is also additive. If the net effect of the products that have an affinity to the regulatory elements of a regulatory unit is negative, the products encoded by this unit’s genes will decrease in concentration, or “degrade”. In addition, the products encoded by regulatory units degrade spontaneously.

The minimal concentration of a product is always 0, but the maximum concentration is different for transcriptional factors (1.0) and diffusive products (10.0). There are two reasons for this 10-fold difference. First, if the maximum concentration of diffusive products was low, it could not be detected in the cells far away from the source cell. Second, when the initial population is formed during simulated evolution (i.e. when the genomes are constructed randomly for the individuals in this population), elements that code for diffusive products are introduced in these genomes less often than elements that code for transcription factors. In contrast to our previous model, the products diffuse here in the body along the filaments that connect the cells (both during development and locomotion). At each time step, the fraction of concentration of a diffusive product transferred between cells is proportional to the difference of concentrations between these cells.

Diffusive products can be considered to be one form of an output produced by a cell (and input received by other cells). Our genome model also includes elements that encode GRN inputs coming from the environment and outputs representing cell actions. These *special elements* are not tied to regulatory units, the graph nodes to which they correspond do not have recurrent connections, and direct connections between input and output nodes are not allowed.

Input elements behave like other regulatory products (transcription factors and morphogens), but their concentration represents an environmental signal. In this paper we use five types of input elements, four of which can be seen as encoding “maternal morphogens”. Three of these morphogens diffuse during development from three point sources, so their perceived concentration in a given cell depends on the current position of this cell. One environmental signal is always present in the same concentration (1.0), throughout development and beyond, when the animat moves (this signal plays the same role as a bias node in artificial neural networks). The fifth chemical starts diffusing in the environment when the animat has finished developing, at which point the animat is supposed to move toward the source of this chemical. Its concentration in each cell depends on the cell’s distance from the source, and goes to 0 for a distance larger than 400 units (noting that the expected value from multiple trials of the initial distance from the center of mass of the animat to the food source is 300).

Whereas input elements encode products whose concentration is determined by the environment, output elements encode products whose concentration impacts the behaviour of the cells and the entire animat after development. In this study we use five output elements representing five possible cell actions: (i) division (when the concentration of the corresponding product crosses a threshold), (ii) rotation to the left and (iii) rotation to the right after division (cell orientation is represented by a vector; the rotation angle depends on the concentration of two products), (iv) contraction and (v) expansion of the filaments linked to the cell (the two products corresponding to these last actions are used only after development, when the mature body is able to move).

### Physics of cell interactions

As in our previous work, the animats are spring-mass systems in which cells correspond to point masses, and neighbouring cells are connected by filaments that act as weightless springs. This neighbourhood relation is determined by calculating the Gabriel graph (Gabriel and Sokal, 1969) of cell positions (Fig. 2).

In our model of two-dimensional swimming, taking after the simulation of undulatory robotic locomotion by Sfakiotakis and Tsakiris (2006), the fluid is stationary and only the spring-edges on the outline of the animat are subject to fluid drag. The force exerted on an edge of length  $L$  is the sum of a tangential component  $F_T = -d_T L v_T^2 \text{sign}(v_T)$  and a normal component  $F_N = -d_N L v_N^2 \text{sign}(v_N)$ , both proportional to the squares of the respective velocity components  $v_T$  and  $v_N$  via fluid drag coefficients  $d_N$  and  $d_T$  (where  $d_N = 200 d_T$ ).

**Soft bodies during development and locomotion** In contrast to our previous work, the rules of physics governing the development and locomotion are identical. These two

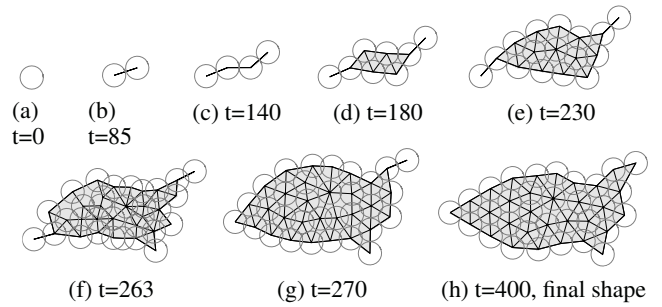


Figure 2: Example of the developmental mechanics (for individual #1, shown in Fig. 3). Cells are represented as circles of radius  $r$  and connected by springs with resting length  $2r$ .

phases remain separate, however, and are different in three respects. First, to prevent excessive forces and movements in the developing embryo due to cell division, cells are slowed down by an extra drag component proportional to the square of their velocity. This correction can be interpreted as the presence of an intracellular fluid more viscous than the external fluid (an alternative, not used here, would be to consider that immature filaments are less stiff). The second difference is that mature filaments in the adult body define polygons that act as pressurized chambers, whose expansion and contraction drive them out of equilibrium and generate a pressure force along the normal of each edge. Thus these chambers constitute a “hydrostatic skeleton” for the animat, which also prevents cells from passing through filaments. Finally, during development cells can break connections or form new ones (as if sprouting filaments or destroying them), whereas the connectivity in the locomoting adult remains fixed.

Formally, this means that the Gabriel graph is recalculated at every step of the development, and each pair of neighbouring cells is connected by a spring whose resting length is the sum of the cells’ radii (in the experiments described here, all cells have the same radius). When a cell produces a daughter cell through division, the new cell is placed closer than the sum of the radii, so the spring that connects these two cells pushes them away from each other, creating a cascading effect in the body. As the organism is growing, cells always attempt to maintain constant distances between them (Fig. 2), and new neighbourhood relations lead to the creation or removal of springs. To keep computational costs reasonable, we used a hard limit of 32 cells.

Once the development is finished, the filaments “mature” i.e. although they retain their elasticity and the body may change shape during movement, the pattern of connections between cells is no longer modified. The initial resting length  $L_0$  of each spring is set to the length it had at the last time step of development. From this point, each cell controls the springs connected to it using the products encoded by two output elements: one product for expansion and one for contraction. The concentrations of expansion

products  $e_1$  and  $e_2$  in the two cells connected by a spring, and the concentrations of contraction products  $c_1$  and  $c_2$  combine additively to modify the resting length according to  $L = (1 + A_{\max}(e_1 + e_2 - c_1 - c_2))L_0$ , where  $A_{\max}$  is a global parameter of the system representing the maximum actuation amplitude ( $A_{\max} = 0.2$  in this paper).

### Evaluation of behaviour and chemotaxis

In our evolutionary model, genetic operators can add elements (duplications), remove elements (deletions), or change the elements' type, sign, and coordinates (point mutations). The first two operations affect the size of the genome and the number of nodes and edges in the GRN, while the change of coordinates affects the affinities between products and regulatory elements. The genetic algorithm is generational, with population size 100, and tournament selection on five randomly drawn individuals. Five of the individuals in each generation are propagated without change to the next generation (elitism), and 20% undergo sexual reproduction (multipoint crossover). An evolutionary run stops when the fitness value is stable over a 500-generation span, which happened between generations #2000 and #3000 in the experiments described here. To accelerate evolution and evaluation, nonviable individuals are removed from the population, where an individual is deemed "viable" if three conditions are met: (i) there is a path between at least one input and the outputs associated with division, contractions or expansions in the GRN, (ii) no cell division happens during the last 100 time steps before the end of development (to allow the physics to equilibrate the adult structure; there is a fixed number of time steps for development), and (iii) the concentrations of expansion or contraction products vary during locomotion. These criteria of viability guide the search for the random genome that will be used to create the genomes of all the individuals in the initial population. These individuals are generated from the seed genome via the duplication, deletion and point mutation operators. The random search of a seed genome requires a few thousand trials.

### Fitness evaluation

After the soft body has fully developed, through cell divisions starting from a single cell, the animat begins to move. In our preliminary experiments we placed a food particle repeatedly at eight random locations forming a circle around the animat's center of mass, and gave higher fitness to animats closer on average to the particle (after a fixed number of time steps). Yet, the evolutionary search in this scenario was not very efficient: only about one third of the evolutionary runs resulted in "champions" that showed some chemotaxis abilities, but at a considerable computational cost due to the required eight test cases for each individual in each generation.

To improve the efficiency of the evolutionary search, we redesigned the fitness function to be composed of five terms

obtained by evaluating an individual in five test situations. (1) The first test situation assessed the ability to move as such: we measured the distance travelled in 10,000 simulation steps. This evaluation stage also allowed to determine the main axis of the animat and its preferred direction of movement. (2) Then, we placed a food particle on the animat's left, between  $-30^\circ$  and  $-90^\circ$  from the main axis, at a distance chosen uniformly and randomly in the range [200,400] from its center of mass (animats cover about 100 units along the main axis), and measured the remaining distance to the particle after 15,000 simulation steps or, if the animat's body overlapped with the particle earlier, the time it took. (3) The third test repeated the second: the state of the animat including its shape and the concentrations of products was reset to the state it had at the end of the first test, and the particle was placed again on the left. (4, 5) The last two tests were similar to tests (2, 3) with the particle placed on the right. The resulting fitness function (maximized by the genetic algorithm) was a linear combination of the distance  $d$  travelled in the first test (via a increasing reward), the remaining distances  $d_n$  from the animat's center of mass to the food particle, and the total durations  $t_n$  of the last four tests (via decreasing rewards):

$$f_{\text{fit}} = \frac{d}{c_m} + \sum_{n=1}^4 \left( 1 - \frac{d_n}{c_f} + s_r \left( 1 - \frac{t_n}{t_{\max}} \right) \right), \quad (1)$$

where  $c_f$  is the maximum distance at which a particle could be placed,  $t_{\max} = 15,000$  is the maximum number of steps in each test,  $s_r$  is the weight of the time reward with respect to the remaining distance reward (here,  $s_r = 4$ ), and  $1/c_m$  is the weight of the distance travelled with respect to the last four tests. This coefficient was set to a value such that, for an individual with efficient locomotion and chemotaxis, the first reward component was of the same order as each of the other four reward components. Our fitness function design promotes the evolution of a simpler behaviour first (here, locomotion), so that a more complex one (chemotaxis) can be built upon it. Considering the relations between learning and evolution, this design brings the fitness function close to a trainer or tutor that promotes gradual development of competences (by "scaffolding" or "shaping" the agent; (Wood et al., 1976; Dorigo and Colombetti, 1994)).

### Results: swimming patterns of four champions from independent evolutionary runs

An analysis of the champions obtained from multiple independent evolutionary runs ( $n = 40$ ) shows that about half of them were able to change direction and to head toward the food particle, while the other half could only swim forward. Our previous work (Joachimczak et al., 2012) had identified four classes of morphologies and styles of motor behavior that emerged more distinctly among the continuum of possible scenarios: symmetrical protrusions on the left

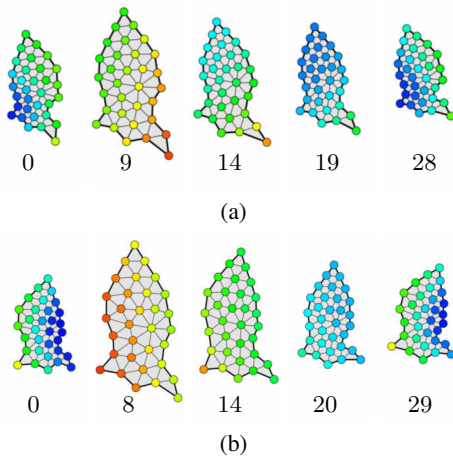


Figure 3: Patterns of cell activation (i.e. concentration of expansion minus contraction products) in individual #1 during one motion cycle while it performs a left (a) or right (b) turn. Red indicates expansion (positive activation); green: resting length; blue: contraction (negative activation). Numbers indicate time steps. The animat swims upward. Videos available at: <http://youtu.be/zi3p164aefY> and <http://youtu.be/Cqt8Fy3CW1A>

and right (or “fins”), a protrusion at the end (or “tail”), undulation of the whole body, and alternation of whole-body pulsations, consisting of either fast expansion and slow contraction of bodies that had a pointy front and a blunt end, or the other way around (fast contraction with a blunt front). Strategies based on pulsations worked by exploiting the fact that fluid drag is proportional to velocity squared. It was also characterized by rapid swings of the concentration of expansion and contraction products in all cells at the same time, whereas in the first three strategies these concentrations varied in a sinusoidal fashion and exhibited phase gradients along the axes of the body. In the present work, it is the pulsation strategy that happened to be the most common among the champions who showed efficient chemotaxis—despite the fact that, in our previous work, individuals with protrusions were the fastest in forward motion. It is interesting to note that the other three strategies also tend to appear in the experiments reported here, but less clearly or only partially as components of a mix (see examples below). This is probably due to the different physics model and the new requirements for chemotactic abilities which could be encouraging pulsations over protrusions or undulation.

We chose four animats among the fittest to be analyzed in greater detail. The pulsation strategy is used by the first three, among which two have elongated bodies in the direction of motion. Individual #1 exhibits a sharp front and a blunt end, contracts slowly and expands quickly (Fig. 3). Individual #2 shows the opposite, with a blunt front and sharp end, contracting quickly and expanding slowly. It also generates thrust by wiggling a “tail” (Fig. 4). Individual #3 also uses a mixed strategy, generating thrust in part from a pul-

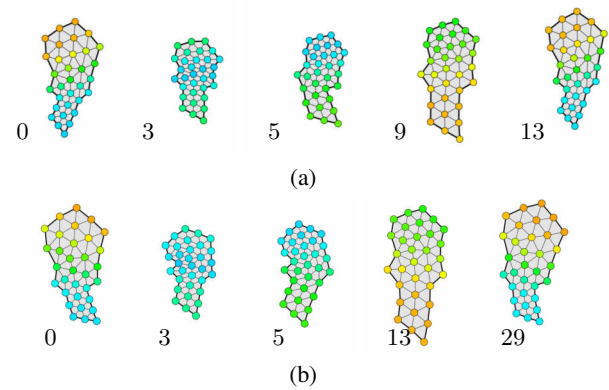


Figure 4: Patterns of cell activation (i.e. concentration of expansion minus contraction products) in individual #2 during one motion cycle while it performs a left (a) or right (b) turn. Red indicates expansion (positive activation); green: resting length; blue: contraction (negative activation). Numbers indicate time steps. The animat swims upward. Videos available at: <http://youtu.be/TS8Q0JfI7o0> and <http://youtu.be/Dw8-YCWodn8>

sating clump in the middle (by fast expansion and slow contraction) and in part by the movement of a small tail, too. A wave of contraction travelling from the front to the back moves this tail in a position perpendicular to the main axis of the animat (equal to the direction of motion) when the animat’s back expands, so that the tail pushes the animat forward (Fig. 5). Individual #4 is sharply different from the other three, as its body is elongated in the direction perpendicular to the main axis, and it moves by using two joined “fins”, which push backward in synchrony to generate a forward movement (Fig. 6). These fins expand when moving backward, and contract on their return. Their motion is induced by a wave of contractions travelling from the back toward the front.

In all four animats the control of chemotaxis performed correctly in the more general situation in which, after the animat reached one food particle, we placed another particle away from the animat without resetting its state to a pre-food situation (although the state was reset during the evaluation phase of the genetic algorithm). When turning toward the food, these four animats did not change their motion pattern, thus it is not immediately obvious how they performed the turn. Defining the level of “activation” of a cell to be the concentration difference between the expansion product and the contraction product, changes in collective activation patterns between the left and right turn can be observed in individuals #1 and #4, while this symmetry is much less clear in individuals #2 and #3 (Figs. 3-6).

To understand how the control of turning worked, we compared the average activation of each cell when food was placed in front of the animat to the activation of each cell when food was placed on the left or on the right. The experiments were performed as follows: (i) the animat was



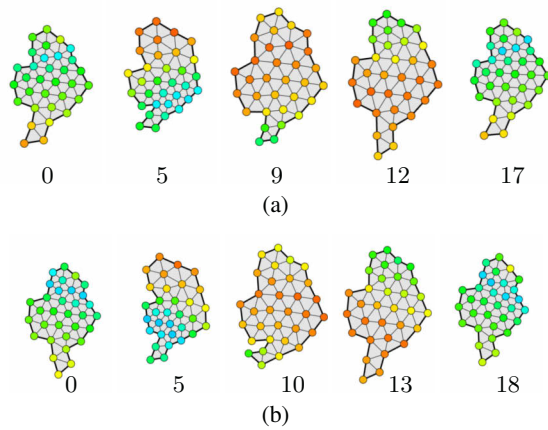


Figure 5: Patterns of cell activation (i.e. concentration of expansion minus contraction products) in individual #3 during one motion cycle while it performs a left (a) or right (b) turn. Red indicates expansion (positive activation); green: resting length; blue: contraction (negative activation). Numbers indicate time steps. The animat swims upward. Videos available at: <http://youtu.be/-HoN7ZGU6W4> and <http://youtu.be/UZeCkWgeA5Q>

allowed to move forward without a particle for 10,000 steps and the direction of movement in the last 50 simulation steps was used to determine its main axis, then the average cell activity was calculated (over 5,000 steps in each case) with (ii) a food particle placed in the front, (iii) 60° to the left, and (iv) 60° to the right. The initial state of the animat (the product concentrations and body shape) in each case was identical to the state at the end of step (i).

Analysis of the changes in the average cell activation indicates that pulsating animats use different strategies based on changing the size of their body parts. Individual #1, the animat that contracts slowly and expands quickly, turns by contracting the part of its body closest to the food (Fig. 7a), as does individual #3, which uses a mixed strategy of pulsation and tail propelled by a wave of contractions (Fig. 7c). On the other hand, individual #2, which contracts quickly and expands slowly, turns by expanding the side opposite to food (Fig. 7b).

In terms of how forces generate motion, when individual #1 with a blunt end expands quickly, it pushes the blunt end against the fluid, and the relative increase in length of the external edges on the right side causes a push toward the left. On the contrary, when individual #2 with a pointy end contracts quickly, a relative increase in edge length on its right side leads to an additional pull toward the left. Although individuals #1-3 showed similar strategies for controlling turns, only in the case of the individual #1, which used a pure pulsation strategy, did we observe an immediate reaction of the cells (i.e. contraction of the springs, but maintaining their pulsation) to a food particle close to them. We also observed similar contractions for cells close to the

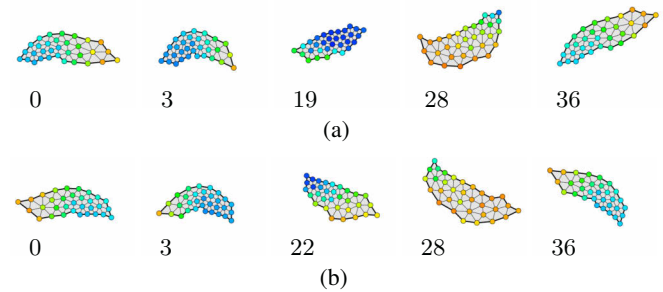


Figure 6: Patterns of cell activation (i.e. concentration of expansion minus contraction products) in individual #4 during one motion cycle while it performs a left (a) or right (b) turn. Red indicates expansion (positive activation); green: resting length; blue: contraction (negative activation). Numbers indicate time steps. The animat swims upward. Video available at: <http://youtu.be/rvM2T8gpDnU>

particle when it was placed inside the body, although this was not experienced during evolution.

In comparison to the first three examples of animats, all propelled by pulsations, individual #4 using two joined fins can turn and move significantly faster, sometimes even overshooting the target but correcting its trajectory afterwards. It is able to switch the direction of the wave of contractions: without food, the wave moves from the back to the front; with a food particle on the right, it moves from the right tip to the left tip, and vice-versa (Fig. 7d). During the switch, the animat maintains the overall motion pattern: the contraction waves are synchronized so that when the right half of the body moves backward, this part contracts. This results in a lower thrust from the right fin, hence a right turn, which is captured by the analysis of average cell activation.

Because the chemical diffusing from the food particle is sensed by all the cells, it is conceivable that the gene network is reacting proportionally to the strength of the incoming signal, and using this direct response to stimulate the turn. To detect this possibility, we computed the Spearman's rank correlation coefficient between the activity (expansion and contraction) of the cells and the distance to the food, i.e. strength of the diffusive signal (Fig. 8). The largest correlations can be observed in individual #1, the pulsating animat that expands quickly. During this animat's behaviour, the high correlations vary in regular patterns. This indicates that cells close to the food expand their springs and cells far from food contract their springs proportionally to the food signal. A similar pattern can often be seen, although not as clearly, in the other two pulsating animats, #2 and #3. This is not, however, the case for the two-finned individual #4. In its case, correlations between activity and distance are relatively low and unstructured, which suggests a more complex strategy during behaviour, including a substantial role played by intercellular communication.

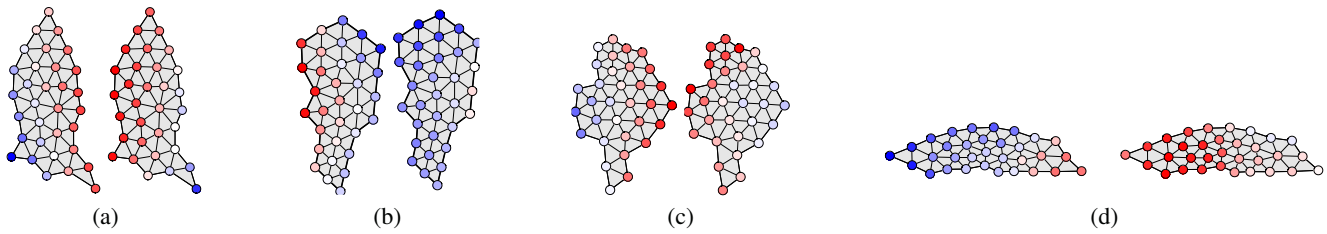


Figure 7: Change in average cell activation (i.e. concentration of expansion minus contraction products) when moving a food particle from the left or right side to the front of an individual. White cell: no change; blue: average activation is lower; red: higher. The four pairs from (a) to (d) correspond respectively to individuals #1 to #4. All animats shown moving upward.

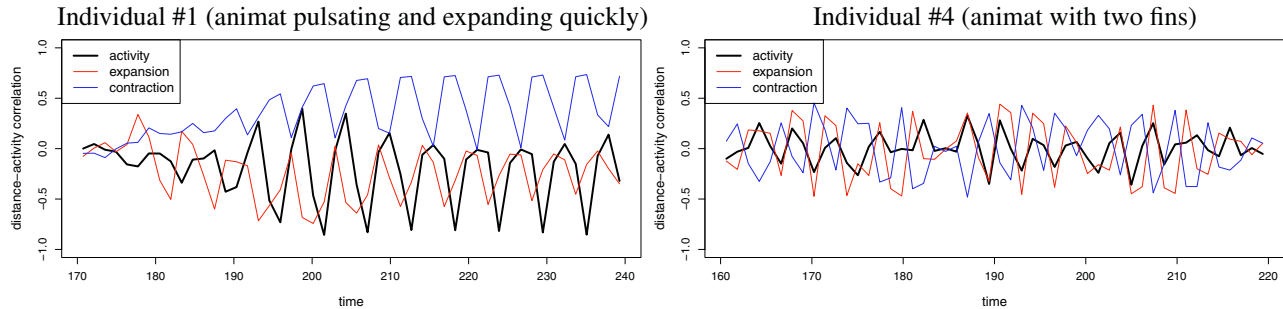


Figure 8: Variations of the Spearman correlation between cell activity and distance to food. Three curves are shown for individuals #1 and #4. They represent the correlation between cell activity  $X$  and the cells' absolute distance to food, where  $X$  stands for the cells' expansion (in red), their contraction (in blue), and their activity (expansion minus contraction, in black). The total period (60 to 70 time steps) corresponds to an animat turning left and moving toward the food. Values near 0 mean no correlation; values near +1 and -1 mean high correlations, which indicate here a direct relationship between the cells' behaviour and their distance to the food.

### Summary, perspective, and future work

In this work, we showed that it was possible for an evolutionary process to successfully produce soft-bodied multicellular animats that can forage for resources in their environment, and whose embryonic development and adult behaviour are both controlled by the same gene regulatory network. Neither a particular morphology nor a particular type of control were enforced, but different strategies were discovered by evolution starting from random genomes. The chemotactic behaviour that emerged from the interplay between the shape of the body and the local responses of differentiated cells, in the absence of any central control, can also be regarded as “minimally cognitive”.

The simulated evolution of a coordinated collective behaviour, where multiple agents are driven by the same distributed controller, provides a way to explore the space of possible morphologies and efficient modes of control in the nascent field of soft-bodied robotics. Our results show that soft robots are able to navigate efficiently and robustly by pulsing the body (symmetrically along the main axis) and expanding their left or right side slightly more in order to turn. We hope that these results can contribute to providing a source of inspiration for the development of new materials and actuators for soft robots.

Before such materials and actuators are available, however, much work can still be accomplished in virtual environments. As future work, we plan to investigate more thoroughly the types of motion and the nature of communication and synchronization among the cells of evolved individuals. In particular, we want to analyze how intercellular communication works to achieve efficient behaviour by exploring a scenario where only a subset of cells (for example, the cells on the surface) can sense the environment, while the rest of the body must rely on indirect information passed through diffusive substances. We would also like to better assess the benefits of distributed control, especially in terms of robustness to damage (e.g. malfunction of springs or cells) and resistance to external disturbances (e.g. distractors and noise).

### Acknowledgments

This work was supported by a PAN-CNRS collaborative project, IO PAN (task IV.3), the Polish National Science Centre (project BIOMERGE, 2011/03/B/ST6/00399), the Tri-city Academic Computer Center (TASK), and the Interdisciplinary Center for Molecular and Mathematical Modeling (ICM, University of Warsaw; project G33-8).

## References

- Banzhaf, W. (2003). On the dynamics of an artificial regulatory network. In *Advances in Artificial Life: Proc. of the 7th European Conference on Artificial Life (ECAL 2003)*, volume 2801 of *LNAI*, pages 217–227. Springer.
- Bongard, J. C. and Pfeifer, R. (2003). Evolving complete agents using artificial ontogeny. In Hara, F. and Pfeifer, R., editors, *Morpho-functional Machines: The New Species*, pages 237–258. Springer Japan.
- Bonner, J. T. (2008). *The Social Amoebae: The Biology of Cellular Slime Molds*. Princeton University Press.
- Dellaert, F. and Beer, A. D. (1996). A developmental model for the evolution of complete autonomous agents. In *From Animals to Animats 4: Proc. of the 4th International Conference on Simulation of Adaptive Behavior (SAB 1996)*, pages 393–401. MIT Press.
- Dorigo, M. and Colombetti, M. (1994). Robot shaping: developing autonomous agents through learning. *Artif. Intell.*, 71(2):321–370.
- Doursat, R. (2008). Organically grown architectures: creating decentralized, autonomous systems by embryomorph engineering. In Würtz, R. P., editor, *Organic computing*. Understanding complex systems, pages 167–199. Springer.
- Doursat, R. (2009). Facilitating evolutionary innovation by developmental modularity and variability. In *Proc. of the 11th Annual Conference on Genetic and Evolutionary computation*, GECCO '09, pages 683–690. ACM.
- Doursat, R., Sanchez, C., Dordea, R., Fourquet, D., and Kowaliw, T. (2012). Embryomorph engineering: Emergent innovation through evolutionary development. In Doursat, R., Sayama, H., and Michel, O., editors, *Morphogenetic Engineering: Toward Programmable Complex Systems*, pages 275–311. Springer-Verlag.
- Eggenberger Hotz, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. In *Proc. of the 4th European Conference on Artificial Life (ECAL 1997)*, pages 205–213. MIT Press.
- Gabriel, K. R. and Sokal, R. R. (1969). A new statistical approach to geographic variation analysis. *Syst. Zool.*, 18(3):259–278.
- Hiller, J. and Lipson, H. (2012). Automatic design and manufacture of soft robots. *IEEE Trans. Robot.*, 28(2):457–466.
- Hornby, G. S. and Pollack, J. B. (2002). Creating high-level components with a generative representation for body-brain evolution. *Artif. Life*, 8(3):223–246.
- Joachimczak, M., Kowaliw, T., Doursat, R., and Wróbel, B. (2012). Brainless bodies: Controlling the development and behavior of multicellular animats by gene regulation and diffusive signals. In *Artificial Life XIII: Proc. of the 13th International Conference on the Simulation and Synthesis of Living Systems*, pages 349–356. MIT Press.
- Joachimczak, M. and Wróbel, B. (2010). Processing signals with evolving artificial gene regulatory networks. In *Artificial Life XII: Proc. of the 12th International Conference on the Simulation and Synthesis of Living Systems*, pages 203–210. MIT Press.
- Joachimczak, M. and Wróbel, B. (2011). Evolving gene regulatory networks for control of artificial cells: signal processing, chemotaxis, multicellular development. In *Proc. of the SynBioCCC Workshop*, pages 4–6.
- Joachimczak, M. and Wróbel, B. (2012). Co-evolution of morphology and control of soft-bodied multicellular animats. In *Proc. of the 14th International Conference on Genetic and Evolutionary Computation*, GECCO '12, pages 561–568. ACM.
- Kowaliw, T., Grogono, P., and Kharm, N. (2004). Bluenome: A novel developmental model of artificial morphogenesis. In *Conference on Genetic and Evolutionary Computation*, GECCO '04, pages 93–104.
- Lopes, R. and Costa, E. (2012). The regulatory network computational device. *Genet. Program. Evol. M.*, 13(3):339–375.
- Meng, Y., Zhang, Y., and Jin, Y. (2011). Autonomous self-reconfiguration of modular robots by evolving a hierarchical mechanochemical model. *IEEE Comput. Intell. Mag.*, 6(1):43–54.
- Nicolau, M., Schoenauer, M., and Banzhaf, W. (2010). Evolving genes to balance a pole. In *EuroGP: 13th European Conference on Genetic Programming*, volume 6021 of *LNCS*, pages 196–207. Springer.
- Rieffel, J., Knox, D., Smith, S., and Trimmer, B. (2013). Growing and evolving soft robots. *Artif. Life*, pages 1–20.
- Schramm, L. and Sendhoff, B. (2011). An animat's cell doctrine. In *ECAL 2011: Proc. of the 11th European Conference on the Synthesis and Simulation of Living Systems*, pages 739–746. MIT Press.
- Sfakiotakis, M. and Tsakiris, D. P. (2006). Simuun : A simulation environment for undulatory locomotion. *Int. J. Model. Simul.*, 26:350–358.
- Shimizu, M., Ishiguro, A., and Kawakatsu, T. (2005). A modular robot that exploits a spontaneous connectivity control mechanism. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1899–1904. IEEE.
- Umedachi, T., Takeda, K., Nakagaki, T., Kobayashi, R., and Ishiguro, A. (2010). Fully decentralized control of a soft-bodied robot inspired by true slime mold. *Biol. Cybern.*, 102(3):261–269.
- van Duijn, M., Keijzer, F., and Franken, D. (2006). Principles of minimal cognition: Casting cognition as sensorimotor coordination. *Adapt. Behav.*, 14(2):157–170.
- Wood, D., Bruner, J. S., and Ross, G. (1976). The role of tutoring in problem solving. *J. Child Psychol. Psych.*, 17(2):89–100.
- Wróbel, B. (2012). Challenges for a-life approach to artificial cognition: in search for hierarchy of cognitive systems. In *Artificial Life XIII: Proc. of the 13th International Conference on the Simulation and Synthesis of Living Systems*, pages 599–600. MIT Press.
- Wróbel, B., Abdelmotaleb, A., and Joachimczak, M. (2012). Evolving spiking neural networks in the greans (gene regulatory evolving artificial networks) platform. In *EvoNet2012: Evolving Networks, from Systems/Synthetic Biology to Computational Neuroscience Workshop at Artificial Life XIII*, pages 19–22.

# Stackelberg-based Coverage Approach in Nonconvex Environments

Bijan Ranjbar-Sahraei, Kateřina Staňková, Karl Tuyls, Gerhard Weiss

Department of Knowledge Engineering

Maastricht University

Email: {b.ranjbarsahraei,k.stankova,k.tuyls,gerhard.weiss}@maastrichtuniversity.nl

## Abstract

This paper introduces StaCo: Stackelberg-based Coverage approach for nonconvex environments. This approach structurally differs from existing methods to cover a nonconvex environment, as it is based on a game-theoretic concept of Stackelberg games. Our key assumption is that one robot can predict (short-term) behavior of other robots. No direct communication takes place among the robots, the approach is decentralized. However, the leading robot can direct the system into the optimal setting much more efficiently just by changing its own position. This paper extends our previous work in which we have introduced the StaCo approach for coverage of a convex environment, with a simpler type of robots. We provide theoretical foundations of the approach. We demonstrate its benefits by means of case studies (using the Sim.Iam software). We show situations in which the StaCo approach outperforms the standard approach, which is based on combination of the Lloyd algorithm and path planning.

**Keywords:** Multi-robot coverage of nonconvex environments, Stackelberg games

## Introduction & Literature Overview

Multi-robot control in an unknown environment is an emerging topic of various research fields (e.g., flocking control (Olfati-Saber, 2006), aggregation (Martinoli et al., 1999), multi-robot coverage (Cortes et al., 2004), formation (Ren and Sorensen, 2008)). This paper focuses on multi-robot coverage.

Most of the proposed solution methods for multi-robot coverage are not applicable in practice, as they encounter difficulties such as failing to find the globally optimal solutions and the inability to account for nonconvex environments (Cortes et al., 2004; Martinoli et al., 1999). As a consequence, despite the wide range of existing works in the domain of multi-robot coverage (Breitenmoser et al., 2010; Butler and Rus, 2004; Cortes et al., 2004; Pimenta et al., 2009; Ranjbar-Sahraei et al., 2012; Schwager et al., 2009), there are still only very few in-field deployments.

Some works dealing with control of the system of multiple robots (not necessarily with multi-robot coverage) have tried to tackle the critical issues of nonconvexity and failing to find the global optimum. Ganguli et al. (2007) solve the distributed Art Gallery Problem in a nonconvex environment. In (Ganguli et al., 2009) the problem of the coordination of a group of robots to achieve rendezvous in a non-

convex environment is treated. An optimal control method to drive a team of multiple robots to target sets under collision avoidance and with proximity constraints in a known environment with obstacles is introduced in (Ayanian and Cumar, 2008). An elegant way of tackling the problem of nonconvex environments is introduced in (Caicedo and Žefran, 2008a,b). The nonconvex region is first transformed by a diffeomorphism to a convex region. Subsequently, the standard Voronoi coverage approach is applied on this region. As the authors themselves state, there is one major drawback of this method: The first phase of this method (transforming the region into a convex region) is computationally very expensive. Moreover, in some cases the solution of the transformed problems does not correspond to the solution of the original problem. Pimenta et al. (2008) apply the geodesic distance measure to Voronoi coverage. While this method is very efficient for some types of environments, it is not guaranteed that the optimal solution will be found for all types of nonconvex regions even if this solution is reachable.

One of the most practically applicable approaches for coverage of a nonconvex environment is introduced in (Breitenmoser et al., 2010). This algorithm combines the standard Lloyd algorithm with a local path planning. However, while this algorithm converges to the locally optimal configuration, it might be extremely slow and does not resolve the issues regarding failure to find the globally optimal configuration.

While the performance of the above mentioned algorithms might be improved via more effective algorithmic implementation, fundamental improvements of the settling time and convergence could be made if the structure of the robotic swarm played a role. Motivated by this idea, this paper introduces a game-theoretic approach which can deal with nonconvexity and local optimality issues more efficiently than the existing algorithms. The Stackelberg Coverage (StaCo) approach is based on the game-theoretic concept of Stackelberg games (Staňková, 2009; Staňková et al., 2013). It assumes that one robot is more advanced than the others. This more advanced robot, called a leader, perceives the environment globally. By its own movement, the leader changes the boundaries of the Voronoi regions of the other robots. Subsequently, and without any direct communica-



tion with the other robots, the leader steers the other robots into a more optimal configuration. The main advantage of the StaCo approach is keeping the benefits of decentralized methods while performing almost as well as the centralized methods with respect to the system optimality; it preserves the simplicity of the major population of the robotic swarm, while one robot can predict behavior of the others and act so that the desired behavior is achieved faster and with a higher precision.

This paper extends the results of (Staňková et al., 2013), where we introduced the StaCo approach for multi-robot coverage of convex environments, toward multi-robot coverage in nonconvex environments. Moreover, more realistic robots are considered in the case studies.

Game theory has been successfully applied in various fields; its known applications in the robotic field relate to pursuit-evasion and search problems (Meng, 2008; Raboin et al., 2010). However, application of the Stackelberg games in the multi-robot coverage of nonconvex environments is new.

In the next sections, we will briefly summarize our previous work and introduce the problem of the Voronoi coverage of a nonconvex environment and its properties. Subsequently, we will explain the StaCo approach in nonconvex coverage and analyze its properties. We will also present case studies in which we demonstrate the advantages of the StaCo approach. We will conclude by discussing the achieved results, limitations of the proposed approach, and the future research directions.

### StaCo in Convex Environments

In (Staňková et al., 2013) we have introduced the StaCo approach for convex environments, as a specific case of a Stackelberg game with one leader (more advanced robot) and multiple followers (very simple robots). We have shown theoretically and by means of case studies that the proposed approach can never perform worse than the standard coverage algorithms, such as the Lloyd algorithm (Cortes et al., 2004), while most of the time the StaCo approach significantly outperforms the standard approaches (by the means of settling time or by finding the globally optimal configuration when standard approaches fail). Figure 1 from (Staňková et al., 2013), illustrates the performance of the StaCo approach in comparison to the classical coverage proposed by Cortes et al. (2004). All experiments are carried out in convex environments and with robotic swarms of different sizes.

### Voronoi Coverage of a Nonconvex Environment

In this section we will informally discuss the problem of nonconvex environment coverage, including a discussion on the existence and the uniqueness of the optimal Voronoi configuration, and uniqueness of this optimal solution.

#### Problem Formulation

The goal is to deploy a group of networked robots in a nonconvex environment, i.e., an environment including

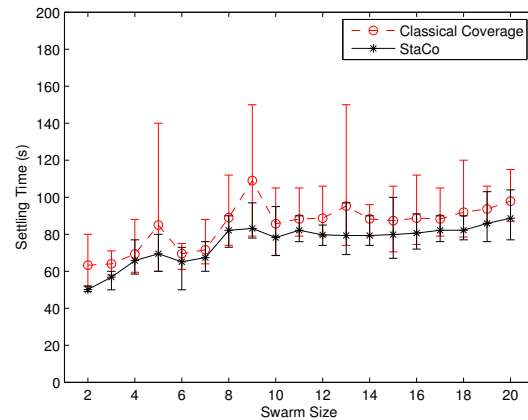


Figure 1: Comparison of the coverage settling time between the proposed StaCo approach and the classical coverage approach for robotic swarms of different sizes in convex environments (Staňková et al., 2013).

free standing obstacles, holes, and/or areas with nonconvex boundaries.

### Problem Properties

**Existence of the optimal Voronoi configuration:** Unlike in a convex environment, the optimal solution does not necessarily exist in a nonconvex environment. This is caused by the fact that the centroids of Voronoi regions are computed in the convex environment, not taking any obstacles into account. However, the centroid of the region might lie on an obstacle or be part of an unreachable region, as shown in Figure 2a. Considering only the situations in which the centroids of the optimally chosen Voronoi regions are reachable, the globally optimal solution exists (but may be impossible to find with standard algorithms).

**Uniqueness of the optimal Voronoi configuration:** The solution configuration does not need to be unique, as there might be multiple solution configurations that are permutations of each other. See Figure 2b for an example of a circular region with a circular obstacle in the middle. Independent of how many multiple robots would be placed in this region, there exist infinitely many optimal Voronoi tessellations in this region.

### StaCo Voronoi Coverage of a Nonconvex Environment

#### Theoretical Foundation & Properties

In this section we formulate multi-robot coverage problem in a nonconvex environment as a dynamic Stackelberg game with one leader and multiple followers, with additional assumptions on the robot's obstacle avoidance behavior. The approach proposed in this section will be referred to as StaCo: Stackelberg-based Coverage Approach. For more details on Stackelberg-based Coverage of convex environments, see (Staňková et al., 2013).

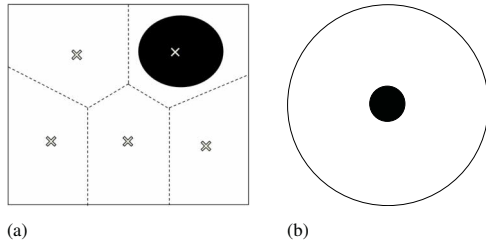


Figure 2: Properties of the Voronoi coverage of a nonconvex environment: (a) Example of the region an obstacle in which the optimal configuration is unreachable. The crosses denote the optimal positions of the robots. (b) Example of a region with (infinitely) many optimal Voronoi tessellations.

Let  $\Omega \subset \mathbb{R}^2$  be a convex region. Let  $n$  convex obstacles  $o_1, \dots, o_n$  be placed in  $\Omega$ . Let us consider  $M$  robots (players) placed in the region at time  $t = 0$  in  $\Omega \setminus \{\bigcup_{j=1}^n o_j\}$ .

One of the players, denoted for the sake of simplicity as player 1, is the *leader*, all other players, denoted by 2,  $\dots$ ,  $M$ , are the *followers*. The roles of the players are assumed to be fixed during the entire duration of the game.

Let  $\mathbf{x}(t) \stackrel{\text{def}}{=} \{x_1(t), x_2(t), \dots, x_M(t)\}$  be the configuration of the robots at time  $t$ , with  $t \in [0, T]$ ,  $\mathbf{x}(0) = \{x_1(0), x_2(0), \dots, x_M(0)\}$  being the initial configuration of the robots and  $\mathbf{x}(T) = \{x_1(T), x_2(T), \dots, x_M(T)\}$  being their final configuration at final time  $T$ , with  $x_i(t) \neq x_j(t)$  if  $i \neq j$ . Note that  $x_i(t) \in \mathbb{R}^2$  for each  $i \in \{1, \dots, M\}$ . Let  $V_i(t)$  indicate the Voronoi region (cell) in which  $i$ -th robot is located at time  $t$ . For each  $\mathbf{x}(t)$  the Voronoi regions are defined by the *Voronoi partition* of  $\Omega$  at time  $t$ ,  $\mathcal{V}(t) = \{V_1(t), \dots, V_M(t)\}$ , generated by the points  $\mathbf{x}(t) = (x_1(t), \dots, x_M(t)) : V_i(t) = \{\omega \in \Omega : \|\omega - x_i(t)\| \leq \|\omega - x_j(t)\|, \forall j \neq i\}$ . System dynamics are given by the following system of ordinary differential equations:

$$\dot{x}_i(t) = u_i(t), \quad i = 1, \dots, M \quad (1)$$

where  $u_i(t) \in \mathbb{R}^2$  is the control (decision) of the  $i$ -th robot at time  $t$ . The cost functions for the leader (robot 1) at time  $t$  is given by

$$C_1(t) = \sum_{i \in \{1, \dots, M\}} \int_{V_i(t)} \|\omega - x_i(t)\|^2 d\omega. \quad (2)$$

Let us assume from now on that  $T$  is defined as the so-called stopping time, i.e., the minimal time such that for each  $\tau > T$  the cost  $C_1(\tau)$  does not change:  $T = \min\{\nu : C_1(\tau) = C_1(\nu) \forall \tau > \nu\}$ . Then the leader minimizes  $C_1(T)$ . Alternatively, the leader might minimize  $T$ . The cost function for the follower  $j \in \{2, \dots, M\}$  at time  $t$  is

$$C_j(t) = \int_{V_j(t)} \|\omega - x_j(t)\|^2 d\omega. \quad (3)$$

The problem of the leader (robot 1) can be then defined as

$$(P_{\text{StaCo}}) \begin{cases} \text{Find } u_1^{(S)}(\cdot) = \arg \min_{u_1(\cdot)} C_1(T), \text{ w.r.t.} \\ u_j(\cdot) = \arg \min_{u_j(\cdot)} \int_{V_j(t)} \|\omega - x_j(t)\|^2 d\omega. \\ \dot{x}_i(t) = u_i(t), \end{cases}$$

with  $j = 2, \dots, N$ ,  $i = 1, \dots, N$ . Note that in a nonconvex case,  $u_1^{(S)}$  involves both obstacle avoidance and reaching the goal behavior. Therefore, the underlying assumption here is that obstacle avoidance is one of the possible controls in (1). Moreover, we want to see how quickly the optimal Voronoi tessellation is found, i.e., the secondary goal is to minimize  $T$ .

**Proposition 1.** *Let at time  $t$  each player  $i$  know only state  $x_i(t)$  and corresponding  $V_i(t)$  and let Hessian of (2) be positive definite at each  $t$ . Then the so-called continuous-time Lloyd descent (Cortes et al., 2004)*

$$u_i^*(t) = \kappa \left( \frac{\int_{V_i(t)} \mathbf{x} d x_i}{\int_{V_i(t)} d x_i} - x_i(t) \right), \quad (4)$$

$\kappa > 0$ , extended by the standard path planning algorithm for obstacle avoidance (Breitenmoser et al., 2010), asymptotically converges to minimal  $C_1(T)$  and to minimal  $C_j(T)$  for  $j = 2, \dots, M$ , provided that the final configuration in which the minimal  $C_1(T)$  is reachable (i.e., no optimal  $x_i$  lies on an obstacle or in an unreachable region).

*Proof.* As shown in (Cortes et al., 2004),  $u_i^*(t)$  defined by (4) with respect to  $\dot{z}_i(t) = u_i(t)$  converges asymptotically to the set of critical points of (2). The critical points of (2) coincide with critical points of (3). If corresponding  $V_i$  is finite, this solution is global due to positive definiteness of (2), as follows from (Du et al., 1999). Assuming that the obstacle avoidance is one of the possible moves in (1) for each robot and that the optimal configuration is reachable from the initial configuration, this concludes the proof.  $\square$

Validation of the positive definiteness of (2) is an open problem (Cortes et al., 2004) and even if the convergence to the global optimum is guaranteed, in general no guarantees on the speed of this convergence exist. This leads us to the question whether there exist algorithms that perform better than the standard Lloyd algorithm (combined with the obstacle avoidance (Breitenmoser et al., 2010) as the covered environment is nonconvex) if we allow the leader (robot 1) to have more information about the state and decisions of the followers.

Note that while a certain position might be unreachable using the classical Lloyd algorithm (a robot might, for example, get stuck on an obstacle, while the Lloyd algorithm would lead the robot to continue through the obstacle), it might be reachable using combination of the Lloyd algorithm and path planning (Breitenmoser et al., 2010). In the reminder of the article, we will refer to the combination of the the Lloyd algorithm and a path planner for obstacle avoidance as the *standard approach*, assuming tacitly that

the StaCo approach uses the same obstacle avoidance mechanisms as the standard approach.

The solution of  $(P_{\text{StaCo}})$  strongly depends on the so-called information pattern, i.e., the amount of information that each player knows and recalls over her own state, the state of the others, and action made by herself and the others during the game (Başar and Olsder, 1999; Staňková and De Schutter, 2011; Staňková et al., 2013). If at each time  $t \in [0, T]$  robot  $P_1$  knows only  $\mathbf{x}(t)$ , the standard approach and the Stackelberg approach might coincide (unless more locally optimal solutions exist). However, if  $P_1$  has more information available, the StaCo approach will perform better than the standard approach (Staňková et al., 2013). The following proposition extends Proposition IV.1. in (Staňková et al., 2013).

**Proposition 2.** *Let player 1 know  $x_j(\tau)$  and  $u_j(\tau)$  (for all  $j \neq 1$ ) for  $\tau \in [t, t + \Delta]$ , with  $\Delta > 0$ , where  $u_j(t)$  is defined by (4). Let  $u_1^{(S)}(t)$  denote the optimal control of player 1, possibly dependent on  $u_j(\tau)$ ,  $\tau \in [t, t + \Delta]$ . Let  $T^{(S)}$ , and  $C_1^{(S)}(T^{(S)})$  denote the corresponding stopping time and the final payoff for player 1 in such a situation, respectively. Then  $C_1^{(S)}(T^{(S)}) \leq C_1^{(L)}(T^{(L)})$ , where  $C_1^{(L)}$  and  $T^{(L)}$  denote the cost of the player 1 if the classical approach, combining the Lloyd algorithm and a path planning, is adopted and the corresponding stopping time, respectively. This inequality holds if the optimal final configuration  $\mathbf{x}(T)$  is reachable from the initial configuration  $\mathbf{x}(0)$ . Moreover, if  $C_1^{(S)}(T^{(S)}) = C_1(T^{(L)})$ , then  $T^{(S)} \leq T^{(L)}$ .*

*Proof.* The leader's decision is not bound by any restrictions. If all past configurations of the StaCo approach and the Lloyd approach coincide, setting the leader's decision to (4) leads to  $T^{(S)} = T^{(L)}$ ,  $C_1^{(S)}(T^{(S)}) = C_1(T^{(L)})$ . Note that the Hessian of (2) might not be positive definite with the leader's decision defined by (4). Thus,  $u_1^{(S)}(t)$  either coincides with (4) when the standard approach is adopted, or, if this choice would lead to only sub-optimal solution,  $u_1^{(S)}(t)$  differs from (4) and leads to a better outcome. This outcome readily follows from Proposition IV.3. in (Staňková et al., 2013).  $\square$

Giving more information to the leader almost always leads to a better outcome for the leader also in a very general setting (Başar and Olsder, 1999; Staňková, 2009), while the StaCo approach never leads to an outcome worse than that reached by standard methods (Cortes et al., 2004; Staňková et al., 2013). This follows from the fact that the classical Lloyd algorithm in which there is no hierarchy among the robots is a special case of the StaCo approach in which the leader does not predict possible position of the other robots and optimizes only locally. Should this behavior be optimal, it would also be adapted by the leader in the StaCo approach.

## Implementation

Following the theoretical description provided in the previous section, in this section we will explain implementation

of the StaCo approach for coverage of nonconvex environments.

In StaCo, the leading robot (we assume that this robot is only one, while keeping in mind that the StaCo approach allows for multiple leaders) has a higher computational capability and more information than the following robots. Subsequently, the overall performance of the system is improved, and the StaCo approach can reach the optimal configuration faster than classical coverage approaches. Moreover, the StaCo approach can also reach the global configuration even if the standard approach fails.

The proposed StaCo approach for nonconvex environments combines three different components. The first one is the Lloyd algorithm, which is already used in the classical Voronoi-based coverage approach (e.g., by Cortes et al. (2004)), and has been mentioned also in the previous section. The second component is the Stackelberg game (described in the previous work of the authors (Staňková et al., 2013)). The third component is a local path planner, including object avoidance and wall following behaviors. This component helps the robot to pass nonconvexities (e.g., obstacles) and move efficiently toward its goal.

**Decision making of leading and following robots:** The leader's prediction of the possible future behavior of other robots and enforcing their optimal behavior via the leader's own movements (without a direct communication) are the main ideas behind StaCo.

The followers follow the simple rules of Lloyd algorithms, as shown in Figure 3a. Each follower continuously computes its Voronoi region center, sets this center as its goal, and tries to reach it, where the goal is a particular point in the 2D-space. Computing the Voronoi center can be done both via having access to global coordination of other robots, or via local communication as proposed by Cortes et al. (2004).

The decision making for the leader is more complex. The leader computes its own movement trajectory efficiently directing the entire group to the best possible configuration. Theoretically, this can be achieved by finding the explicit solution of  $P_{\text{StaCo}}$ , introduced in previous section. However, computation of such a solution analytically is very complicated and therefore, we compute the approximate solution of  $P_{\text{StaCo}}$  in a numerical way. In this numerical computation, the leader predicts possible behavior of the other robots as a response to its own behavior only for a fixed time interval and fixed number of directions for the leader's next move. The direction which implies the minimal cost function value is chosen as the immediate leader's goal. The immediate goal will be updated by the same procedure after the a priori fixed time. The sequence of such short-term goals defines the leader's movement trajectory. See Figure 3b for the scheme of the leader's behavior.

**Remark 3.** Note that the leader's prediction quality is directly influenced by the type and amount of information that is available to the leader. In our experiments, it is assumed that the leader knows the position of other robots and knows their dynamics. Additionally, the leader knows the map of

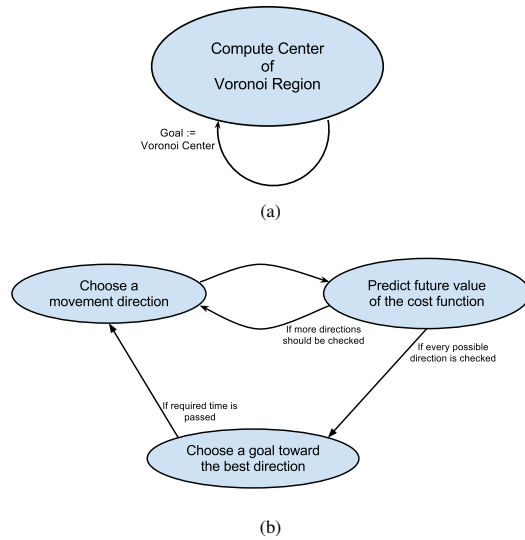


Figure 3: Scheme of robot behaviors/decisions: (a) simple behavior of a follower robot, in which the goal is a specific location in 2D space that the robot tries to reach using a local path planner. (b) the decision making of a leader robot in which the leader figures out which movement direction concludes to the best overall performance of the group.

the environment (more precisely, the position of obstacles). As long as only one robot or a very few number of robots has these abilities among a large number of simple robots, we can consider these assumptions as being practical and feasible, especially due to high capabilities of most modern robots.

**Remark 4.** Solving the  $P_{StaCo}$  problem in a numerical way can be improved in different aspects. First of all, the more movement directions the leader considers, the more accurate the movement, and the higher efficiency. Any metaheuristic which helps the robot in finding the best movement direction can be incorporated into the proposed approach. For example, with use of the A\* search algorithm one can penalize choosing a trajectory passing very close to the obstacles, as opposed to an obstacle-free trajectory. Many other search heuristics (e.g., GA and Simulated Annealing) can be used to find the best movement directions in the fast and accurate manner (Resende and de Sousa, 2004). However, study of these techniques is beyond the main scope of this paper, which focuses on overall applicability and efficiency of StaCo.

**Local path planner design:** Local path planner has an important role in adapting the StaCo approach for convex environment, proposed in previous work of the authors (Stařková et al., 2013), to coverage of nonconvex environments. Different local path planners are available for autonomous robots (Buniyamin et al., 2011). In (Breitenmoser et al., 2010) the *TangentBug* planner was used to tackle environment nonconvexities. In this paper, we use the hybrid controller proposed by Egerstedt (2000), in which robots follow three basic behaviors of *go-to-goal*, *avoid-obstacle*,

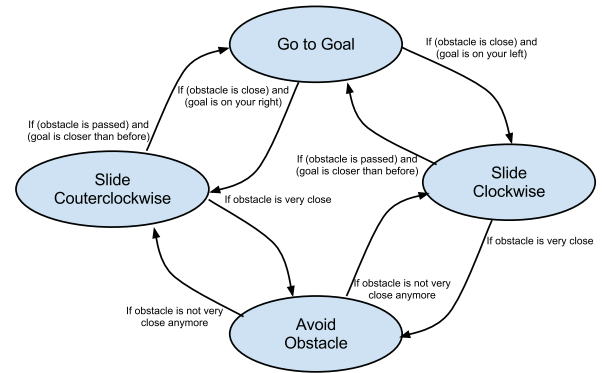


Figure 4: Hybrid automata used as local path planner designed for local path planning in nonconvex environments.

and *sliding along walls*. The hybrid automaton for this behavior-based path planner is shown in Figure 4, where the transitions and resetting values are explained qualitatively. We have adopted this path planner, as it is widely used by other robotic researchers, due to its simplicity of implementation, its robustness to environment changes, and its efficiency in finding the best available trajectory avoiding the obstacles of different shapes. Interested readers are referred to (Egerstedt, 2000) for more details on the design of this local path planner.

While the robot is moving toward its goal (i.e., Voronoi region center), if an obstacle appears in its way, the robot slides on the surface of the obstacle, until it gets to a position from which the goal is closer than it was before detecting the obstacle (i.e., the obstacle is already passed), then it switches back to the standard goal following. If at some point the robot gets very close to the obstacle, a pure repulsive behavior emerges which avoids collision with the obstacle.

**Remark 5.** The hybrid controller used for local path planning is always able to pass obstacles and move toward the reachable goals (Egerstedt, 2000). However, when the goal is unreachable (i.e., inside of an obstacle or bounded by obstacles), the path planner keeps moving the robot along the borders of the obstacle. In this paper we let the robot move around the obstacle as far as it is required, which will make an average position closer to the goal. However, a useful alternative is to stop the robot after one full cycle around the obstacle, as proposed by Breitenmoser et al. (2010).

## Simulations

In this section, we will study the performance of the proposed StaCo approach in comparison to the classical Voronoi-based coverage approach in nonconvex environments introduced by Breitenmoser et al. (2010). Firstly, the simulation environment and the mobile robot platform will be introduced. Secondly, the efficiency of the StaCo approach compared to the classical approach will be illustrated in two case studies.



## Simulation Setting

For examining the performance of the proposed coverage approach, Sim.I.am, a MATLAB-based educational software developed by de la Croix and Egerstedt (2013), is used in different coverage scenarios. The mobile robot platform, which is implemented in Sim.I.am, is the Khepera III (K3). The K3 is equipped with 11 infrared (IR) range sensors, nine of which are located in a ring around the robot and two are located on the underside of the robot. The IR sensors are complemented by a set of five ultrasonic sensors.

In the previous work of the authors (Staňková et al., 2013) the simulations of the proposed approach were carried out with a group of mass-less robots (i.e., neither the dynamic nor the kinematic model of the real-world robots were considered). In contrast, in the new simulator the robots (the nonholonomic Khepera robots) are much more realistic. Therefore, compared to (Staňková et al., 2013) and vast majority of papers on the Voronoi coverage, simulations in this paper are much closer to the real-world scenarios.

In the simulation environment, we have access to the array of nine IR sensors that encompass the K3. IR range sensors are effective in the range  $0.02\text{ m}$  to  $0.2\text{ m}$  only. Since the K3 has a differential wheel drive, it has to be controlled by specifying the angular velocities of the right and left wheels ( $v_l, v_r$ ). Therefore, the conversion between a unicycle input, the forward and angular speeds, to differentially driven inputs are implemented based on following equation for the  $i$ th robot:

$$\begin{aligned} \dot{x}_i^1 &= R(v_l + v_r) \cos(\theta) \\ \dot{x}_i^2 &= R(v_l + v_r) \sin(\theta) \\ \dot{\theta}_i &= R(v_r - v_l)L \end{aligned} \quad (5)$$

where  $x_i^1$  and  $x_i^2$  denote coordination of the robot in horizontal and vertical directions,  $R$  is the radius of the wheels, and  $L$  is the distance between the wheels, which are known a priori. Wheel encoders are used to provide required information to the odometry of robot. The relevant information needed for odometry is the radius of the wheel, the distance between the wheels, and the number of ticks per full turn of the wheel, which are all implemented internally in the simulator. Note that the equations (5) extend equation (4) in which the robot is considered to have no mass and to be holonomic.

The embedded controller described previously in the form of a hybrid automaton (Figure 4) is used to deal with the local path planning tasks. The transition for moving from “go to goal” behavior to the “sliding mode” happens when a robot is in a distance less than 15 cm, and it will move to the pure repulsive behavior (i.e., obstacle avoidance) when the robot is closer than 6 cm to the obstacle.

The prediction time in which the leading robot finds the approximate best movement direction (Figure 3) is a period of 3 seconds and the robot calculates the final value for moving to 8 different directions (i.e., right, up-right, up, up-left, ..., down right) for this period of time. Note that the way in which the leading robot computes its next step agrees with

the concept of the model predictive control known from the optimal control theory literature (Mayne et al., 2000).

## Results

Efficient coverage behavior of StaCo in convex environments is reported in our previous work (Staňková et al., 2013): As shown in Figure 1, the StaCo approach outperforms the classical coverage approaches in most of the environmental settings, and in the worst case StaCo and classical techniques have equal performance.

We use two case studies to show the high performance of StaCo in coverage of nonconvex environments and simultaneously we compare the results with the approach proposed by Breitenmoser et al. (2010).

The two non-trivial case studies for examining the StaCo approach in nonconvex environments are illustrated in Figure 5. In both scenarios five robots are initiated at random positions. The obstacles make the environment nonconvex which consequently makes an efficient coverage difficult.

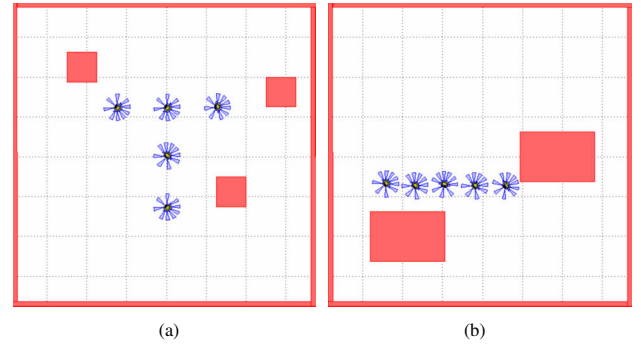


Figure 5: Initial settings for two experiments with five robots: (a) Scenario I. (b) Scenario II.

In both scenarios (Figures 5a and 5b), first the classical coverage approach for nonconvex environments is applied. In this approach, robots move toward their goals while a local path planner is used for obstacle avoidance and obstacle following purposes. Afterwards, we apply the StaCo approach to the same initial configurations, where one (randomly selected) robot acts as the leader. In Figure 5a, the robot in the center and in Figure 5b, the leftmost robot are the leaders. As explained earlier, the leader enforces its decisions on the other robots via its movements in the environment.

The coverage results of initial configurations shown in Figs. 5a and 5b are shown in Figures 6a-6c and Figures 6d-6e, respectively. Firstly, the robot trajectories for both classical coverage and StaCo approaches are shown (Figures 6a and 6d). Subsequently, the final configuration and the final Voronoi tessellation for each approach is illustrated (Figures 6b and 6e). Finally, the cost functions (2) for both approaches are plotted in Figures 6c-6f with respect to time.

As shown in Figures 6c and 6f, the StaCo approach finds the optimal configuration in a short time, while the classical approach is unable to find this optimal configuration

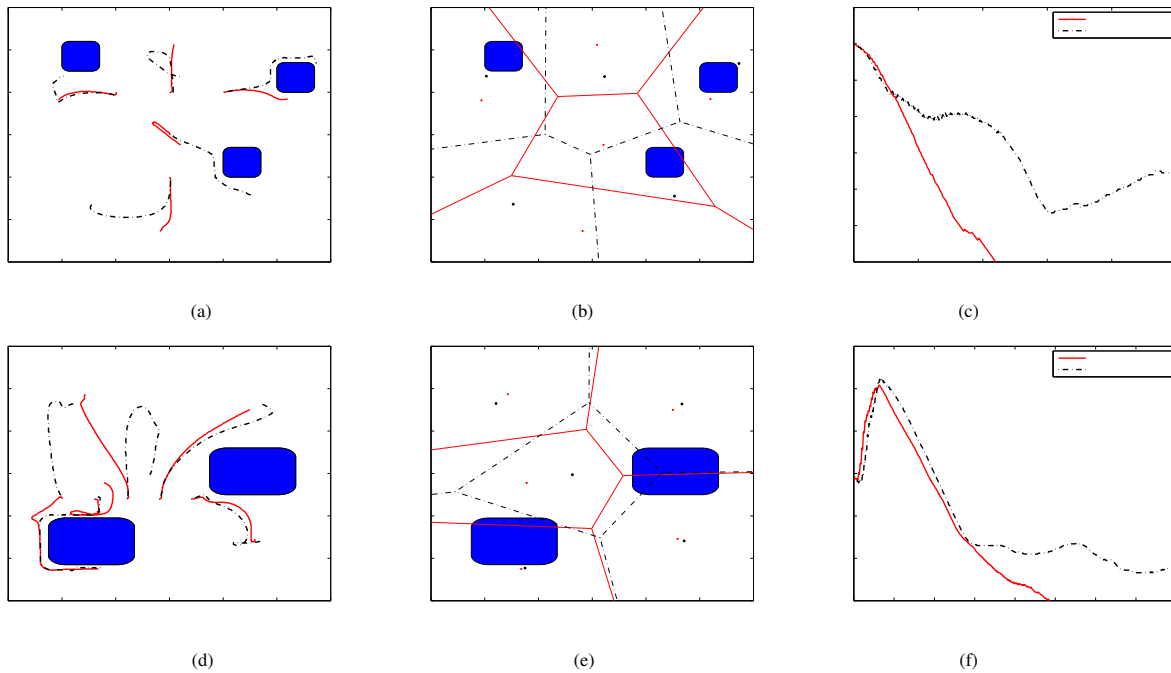


Figure 6: Experimental comparison between classical coverage approach (dash-dotted line) and StaCo approach (continuous line): (a) robot trajectories for Scenario I. (b) final robot configurations and final Voronoi tessellations for Scenario I. (c) cost function comparison for Scenario I. (d) robot trajectories for Scenario II. (e) final robot configurations and final Voronoi tessellations for for Scenario II. (f) cost function comparison for Scenario II.

even in the long term. From the trajectories of the individual robots (Figures 6a and 6d) it can be seen that while the classical approach moves the robots toward the optimal positions blindly (they can not predict whether obstacles will cause problems or not), in the StaCo approach, the leader which has access to more information and higher computation abilities, can enforce the entire group to move to the optimal configuration more efficiently.

### Discussion, Conclusions & Future Research

In this paper we have shown the high potential of the StaCo approach in the coverage of a nonconvex environment. In the situations in which the leader can predict the long-term behavior of the other robots, the StaCo approach outperforms the standard approaches for coverage of nonconvex environments. Moreover, we have shown that the StaCo approach outperforms the standard approaches even if the prediction capabilities of the leading robot are very limited. Extending the leader's prediction horizon will then lead to even better results. The main advantage of StaCo compared to any possible centralised coverage approach, is that StaCo does not rely on any direct communication between robots.

Leader's predictions might become computationally expensive especially in an environment with many obstacles and/or if the robotic swarm is very large. More advanced optimization methods might then have to be applied to overcome this possible drawback. Our next research step is to address this issue.

Moreover, we plan to implement StaCo in a real-robot setting using a combination of TurtleBots as the leaders and e-pucks as the followers. Although we do not expect implementation problems due to the available advanced robots, we need to explore in detail the level of StaCo precision that can be achieved in in-field scenarios that have different environmental and technical conditions.

Last but not least, our future research will include expressing the optimal leader's behavior in explicit form and extending the number of leaders (in such a case the hierarchy between the individual leaders might play a role).

### References

- Ayanian, N. and Cumar, V. (2008). Decentralized feedback controllers for multi-agent teams in environments with obstacles. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, pages 19 – 23.
- Başar, T. and Olsder, G. J. (1999). *Dynamic Noncooperative Game Theory*. SIAM, Philadelphia, Pennsylvania.
- Breitenmoser, A., Schwager, M., Metzger, J., Siegwart, R., and Rus, D. (2010). Voronoi coverage of non-convex environments with a group of networked robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4982 – 4989.
- Buniyamin, N., Wan Ngah, W., Sariff, N., and Mohamad, Z. (2011). A simple local path planning algorithm for

- autonomous mobile robots. *International Journal of Systems Applications, Engineering & Development*, 5(2):151 – 159.
- Butler, Z. and Rus, D. (2004). Controlling mobile sensors for monitoring events with coverage constraints. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1568 – 1573.
- Caicedo, M. and Žefran, M. (2008a). A coverage algorithm for the class of nonconvex regions. In *Proceedings of the 2008 IEEE International Conference on Decision and Control*, pages 4244 – 4249.
- Caicedo, M. and Žefran, M. (2008b). Performing coverage of non-convex domains. In *Proceedings of the 2008 IEEE International Conference on Systems & Control*, pages 1019 – 1024.
- Cortes, J., Martinez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243 – 255.
- de la Croix, J.-P. and Egerstedt, M. (2013). Sim.I.am. <http://gritslab.gatech.edu/projects/robot-simulator/>. [Online; accessed April-2013].
- Du, Q., Faber, V., and Gunzburger, M. (1999). Centroidal Voronoi tessellations: Applications and Algorithms. *SIAM Review*, 41(4):637 – 676.
- Egerstedt, M. (2000). Behavior based robotics using hybrid automata. In *Hybrid Systems: computation and control*, pages 103 – 116. Springer Berlin Heidelberg.
- Ganguli, A., Cortes, J., and Bullo, F. (2007). Distributed coverage of nonconvex environments. In Saligrama, V., editor, *Networked Sensing Information and Control*, pages 289 – 305. Springer US.
- Ganguli, A., Cortes, J., and Bullo, F. (2009). Multirobot rendezvous with visibility sensors in nonconvex environments. *Robotics, IEEE Transactions on*, 25(2):340 – 352.
- Martinoli, A., Ijspeert, A., and Gambardella, L. (1999). A probabilistic model for understanding and comparing collective aggregation mechanisms. In *Proceedings of the Fifth European Conference on Artificial Life (ECAL99)*, pages 575 – 584.
- Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814.
- Meng, Y. (2008). A game-theory based multi-robot search approach for multiple targets. *IEEE Robotics & Automation Magazine*, 5(4):341 – 350.
- Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: Algorithms and Theory. *IEEE Transactions on Automatic Control*, 51(3):401 – 420.
- Pimenta, L. C. A., Kumar, V., Mesquita, R., and Pereira, G. A. S. (2008). Sensing and coverage for a network of heterogeneous robots. In *Decision and Control (CDC). 47th IEEE Conference on*, pages 3947 – 3952.
- Pimenta, L. C. A., Schwager, M., Lindsey, Q., Kumar, V., Rus, D., Mesquita, R. C., and Pereira, G. A. S. (2009). Simultaneous coverage and tracking (SCAT) of moving targets with robot networks. In *Proceedings of the Eighth International Workshop on the Algorithmic Foundations of Robotics (WAFR 08)*, pages 85 – 99.
- Raboin, E., Nau, D., Kuter, U., Gupta, S. K., and Svec, P. (2010). Strategy generation in multi-agent imperfect-information pursuit games. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 947 – 954.
- Ranjbar-Sahraei, B., Weiss, G., and Nakisaee, A. (2012). A multi-robot coverage approach based on stigmergic communication. In *Multiagent System Technologies*, volume 7598 of *Lecture Notes in Computer Science*, pages 126 – 138. Springer.
- Ren, W. and Sorensen, N. (2008). Distributed coordination architecture for multi-robot formation control. *Robotics and Autonomous Systems*, 56(4):324 – 333.
- Resende, M. G. C. and de Sousa, J. P., editors (2004). *Metaheuristics: Computer Decision-Making (Applied Optimization)*. Springer.
- Schwager, M., Rus, D., and Slotine, J. J. (2009). Decentralized, adaptive coverage control for networked robots. *International Journal of Robotics Research*, 28(3):357 – 375.
- Staňková, K. and De Schutter, B. (2011). Stackelberg equilibria for discrete-time dynamic games – Part I: Deterministic games. In *Proceedings of the 2011 IEEE International Conference on Networking, Sensing and Control*, pages 249 – 254, Delft, The Netherlands.
- Staňková, K. (2009). *On Stackelberg and Inverse Stackelberg Games & Their Applications in the Optimal Toll Design Problem, the Energy Market Liberalization Problem, and in the Theory of Incentives*. PhD thesis, Delft University of Technology, Delft, The Netherlands.
- Staňková, K., Ranjbar-Sahraei, B., Weiss, G., and Tuyls, K. (2013). StaCo: Stackelberg-based coverage approach in robotic swarms. In *Proceedings of ADAPTIVE 2013*, pages 71–76.

# Simulating Limited Diversity in Evolution of Influenza

Takahiro Sasaki

Sony Computer Science Laboratories, Tokyo, Japan  
sasaki@csl.sony.co.jp

## Abstract

In many cases, analyses on infectious diseases focus on how the epidemic arises, spreads, and whether diminishes or gets fixated among host populations under particular conditions, without taking the evolutionary perspective into account. With some infectious diseases, however, the pathogens themselves evolve comparatively rapidly during the time course, so that the co-evolutionary dynamics among hosts and pathogens should be considered at the same time. In this paper, we focus on influenza and propose a bilayered, multi-agent-based simulation that combines an epidemic model and a viral evolution model. The latter model includes genomic segments of the viruses whose evolutionary paths are guided by two selective pressures; one originates from the viral-host immune interaction and the other originates from intra-genomic constraints within the virus. By including such a micro-level representation in the model, we show mechanisms that generate the limited diversity of viruses, which is a fundamental yet unexplained temporal characteristic observed in the evolution of influenza. The full version of this work has already been published in (Sasaki, 2013).

## Introduction: Limited Diversity

While influenza is a quite common infectious disease, the pattern of its global circulation and evolutionary dynamics still pose many questions to be answered (Nelson and Holmes, 2007). *Limited diversity* observed in the viral evolution of influenza is the one that has not been fully investigated. Influenza viruses have single-stranded RNA, which lacks an error correction mechanism and thus results in the high mutation rate of the genome. Thus, while continual changes accumulate in the viruses as time passes, there is always a chance for a new genetic lineage to diverge, which could theoretically result in ever-increasing explosive diversity. However, in reality, the diversity of the existing viruses is relatively limited to a certain extent at the population level and at any particular point in time. On one hand, this evolutionary behavior is in practice an important matter to be considered when we make effective plans and take medical action for minimizing the negative impact of infectious diseases, while on the other hand, it purely engages our inquisitive minds and makes us think about what causes the emergence of limited diversity in influenza viruses.

For example, Ferguson et al. (2003) discussed the role of short-lived non-specific cross-immunity as a possible mechanism that limits viral diversity. However, the biological mechanisms or evidence that actually support it have not necessarily been fully shown. By adding a more precise micro-level representation of viral evolution to the model, we propose another mechanism that also generates the limited diversity with more natural explanation than the previous work.

## Methods: Model

We propose a bilayered multi-agent-based model that consists of an epidemic circulation layer and a viral evolution layer. As in typical epidemic simulations, each member of the host population is individually represented as a virtual agent (Eubank et al., 2004; Parker, 2007; Epstein, 2009). The viruses, also represented as agents, are replicated and circulated among the hosts with their genomes evolving.

Reflecting the fact that the genome of the real influenza virus is composed of eight distinct but possibly interrelated segments, we represent our pseudo virus as a composite of multiple strings instead of treating it as just a monolithic component like most of the previous studies have done. In our model, for simplification we just consider two strings. The first, string  $g$ , determines the epitope recognized by the immune systems of the hosts, and the other, string  $r$ , does not directly interact with the immune systems but mutually regulates the evolution of  $g$  and of itself.

The key and unique point of our model is that the evolution of viruses is driven and/or regulated by two distinct selective pressures (Fig. 1). On one side, string  $g$  is subjected to repulsive pressure because immune systems prevent hosts from being infected repeatedly by viruses with the same genomic configuration. Thus,  $p_1$ , a component of viral fitness derived from the repulsive pressure, can be represented as a monotonic increasing (possibly sigmoidal) function of  $d_i$ , the distance between the binary pattern of  $g$  and the binary pattern of those held in immune memory. On the other side, however,  $g$  is subjected to attractive pressure because of intra-genomic constraints. Even though the viral genome



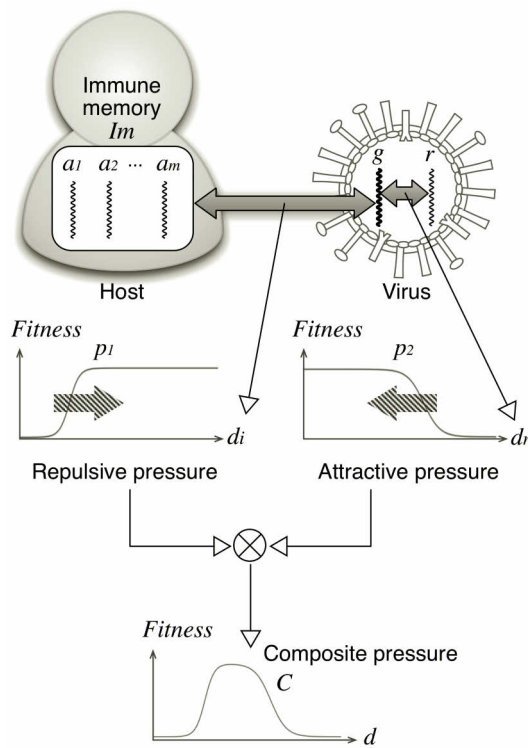


Figure 1: Micro-model of viral evolution

is separated into several segments, none of these segments can evolve totally independently from the others but a certain degree of consistency among them must be held to keep viral functions effective (Rambaut et al., 2008). Thus,  $p_2$ , another component of viral fitness derived from the attractive pressure, can be represented as a monotonic decreasing function of  $d_r$ , the distance between the epitope segment  $g$  and regulatory segment  $r$ .

As a result of these two selective pressures working in opposite directions, the shape of the composite pressure  $C = p_1 \times p_2$  becomes bell-shaped. The curve indicates the existence of a *window of diversity* that determines a certain advantageous range for viruses to thrive, i.e., “*Move a certain distance from here but not too far.*”

The details of our model have already been described in (Sasaki, 2013).

## Experimental Results

We conducted experiments with the model described in the previous section to investigate its dynamical behavior. Figure 2 shows the changes in diversity, which were simply measured as the number of distinct types of viruses co-circulating in the world in each time step. The dashed line shows that when the viruses evolved under only the selective pressure of immune interaction, the diversity of the viral population grew rapidly and explosively as the steps of the simulation proceeded. In contrast, the solid line shows

that when the viral evolution was under combined selective pressure from both immunity and intra-genomic constraints, the diversity did not grow explosively but saturated at a certain limit, which in this case was around 50, even in the later steps. The result indicates that our hypothetical mixture of two counter-directed selective pressures could be a possible factor to drive the evolutionary changes but at the same time limit its diversity.

Reproducing limited diversity with the simulation model is not our ultimate objective, but it is an inevitable challenge that needs to be tackled to explore the entire spectrum of temporal evolutionary behavior of the systems.

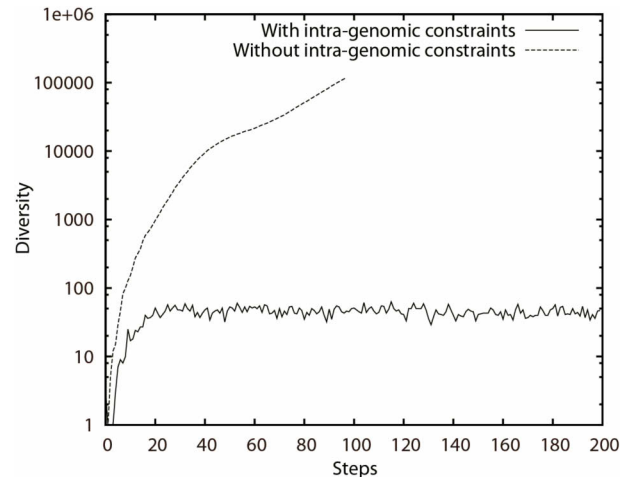


Figure 2: Experimental results

## References

- Epstein, J. M. (2009). “Modelling to contain pandemics”. *Nature*, 460:687.
- Eubank, S., Guclu, H., Kumar, V. S. A., Marten, Srinivasan, A., Toroczkai, Z., and Wang, N. (2004). “Modelling disease outbreaks in realistic urban social networks”. *Nature*, 429:180–184.
- Ferguson, N. M., Galvani, A. P., and Bush, R. M. (2003). “Ecological and Immunological determinants of Influenza evolution”. *Nature*, 422(6930):428–433.
- Nelson, M. I. and Holmes, E. C. (2007). “The evolution of epidemic influenza”. *Nature Reviews Genetics*, 8(3):196–205.
- Parker, J. (2007). “A Flexible, Large-Scale, Distributed Agent Based Epidemic Model”. In *WSC '07: Winter Simulation Conference*, pages 1543–1547.
- Rambaut, A., Pybus, O. G., Nelson, M. I., Viboud, C., Taubenberger, J. K., and Holmes, E. C. (2008). “The genomic and epidemiological dynamics of human influenza A virus”. *Nature*, 453(7195):615–619.
- Sasaki, T. (2013). Hierarchical Multi-Agent-Based Model for Simulating the Prevalence and Evolution of Influenza Virus. In *Spring Simulation Multi-Conference 2013*, pages 62–69.

# The Origin of Money: An Agent-Based Model

Timothy A. Moran<sup>1</sup>, Markus Brede<sup>1</sup>, Antonella Ianni<sup>1</sup> and Jason Noble<sup>1</sup>

<sup>1</sup>The University of Southampton, Southampton SO17 1BJ, UK  
tam1g09@soton.ac.uk

## Abstract

The benefits of money as a medium of exchange are obvious, but the historical origin of money is less clear. An existing economic model of monetary search is reproduced as an agent-based simulation and an evolutionary algorithm is used to model social learning. This approach captures the way in which different equilibria can arise, including solutions in which one or two goods come to be used as money. In the case where monetary goods have identical properties, multiple equilibria can be reached with a dependence on the starting beliefs of agents. In our analysis we also consider the evolutionary dynamics that allow for a small chance of mutations in strategies. In some cases our findings show evolutionary paths by which use of particular monetary goods can collapse.

## Introduction

The economy is a complex adaptive system (Beinhocker, 2007). Money and its general acceptance as a medium of exchange lie at the heart of most economic activity. Its use offers a convenient alternative to barter, allowing agents who share a belief in its acceptability to trade indirectly using a monetary good that offers them no direct utility. It also offers a decentralised alternative to personal credit arrangements if the acceptance of the money is widespread.

But the value of money as a medium of exchange only arises if that money is widely accepted. The initial growth in the acceptance of money involves the reinforcement of agent beliefs from repeated successful transactions with an emergent form of money, and does not require any centralised coordination. Building an agent-based model of such a system will allow us to assess the plausibility of different historical pathways to the emergence of money, and also to study the conditions that lead to a collapse in the acceptance of a particular monetary system, a topic that economic models have so far neglected.

This paper begins by introducing an economic search model of money and its use in experiments with real and artificial agents. This model is then implemented as an agent-based simulation and extended to allow agents to learn successful trading strategies. Evolutionary paths towards the Nash equilibria are shown.

## A Search Model of Money

Kiyotaki & Wright (1989) proposed a probabilistic search and matching model that can support monetary equilibria where

useful commodities are valued as media of exchange. The economy consists of three types of agent (I, II and III) who can each hold a single unit of one of three goods (1, 2 and 3).

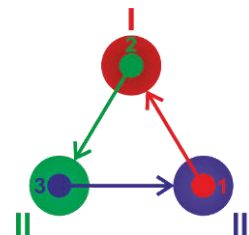


Fig. 1. Production and consumption in the Kiyotaki-Wright model. Type I agents consume good 1 and produce good 2, type II agents consume good 2 and produce good 3, and type III agents consume good 3 and produce good 1

Agents can produce one type of good, but only derive utility by consuming a different type of good. An agent will consume its consumption good immediately, and will produce its production good after consuming. (Thus an agent is never empty-handed.) Since no agent produces its own consumption good, inter-agent trade is necessary for agents to derive utility.

Agents have the opportunity to trade through a random matching process. In every time period, agents are randomly paired and given the opportunity to trade. The model is designed to ensure that there exists no ‘double coincidence of wants’ (Jevons, 1875) between any two agents. In other words, for trade to take place at least one agent must be willing to accept a good other than its consumption good. (This sets the stage for a good to potentially emerge as a medium of exchange.) Trade only takes place when both agents in a pair value their partner’s holding more highly than their own. Thus agents will always accept their own consumption good and they will never trade with an agent holding the same good that they are already holding.

Trade in other goods depends on the trading strategies of agents. To differentiate between the good types, the model imposes different storage costs for each. Letting  $c_j$  denote the cost of holding good type  $j$  between trading turns, then  $c_3 > c_2 > c_1$ , meaning that good 3 is the most costly to store and good 1 is the least costly.

Agents attempt to maximise their expected discounted lifetime utility. If they do not believe that any particular good will increase their chance of trading in a subsequent turn then they consider only the physical properties of the goods, and

will only accept their consumption good or a commodity that is cheaper to store than their current holding. In this **fundamental equilibrium** type I and type III agents will never trade directly, as type I agents aim to minimise costs by never accepting good 3 from type II agents. In a sense, type II agents are willing to use good 1 as money, but only because it is cheaper to store than their production good (3).

As Duffy (2001) points out: ‘An agent *speculates* when he accepts a good in trade that is more costly to store than the good he is currently storing with the expectation that this more costly-to-store good will enable him to more quickly trade for the good he desires to consume.’ For a sufficiently high utility of consumption (or, equivalently, sufficiently low storage costs) type I agents are willing to accept good 3 from type II agents, allowing them to subsequently trade directly with type III agents for their consumption good. In this case a **speculative equilibrium** is supported; type I agents are now willing to use good 3 as money, even though it costs more to store than their production good (2).

In general the trading strategies for any type of agent can be summarised in the form  $a > b > c$ , meaning that  $a$  is that agent’s favourite good and  $c$  is that agent’s least favourite good. The agent will trade any holding in exchange for good  $a$  (the agent’s consumption good), will trade holding  $b$  only in exchange for good  $a$ , and will trade holding  $c$  in exchange for any other good. The specific trading strategies of the three types of agents used in this model are shown in Fig. 2.

Equilibrium	Type I	Type II	Type III
Fundamental	$1 > 2 > 3$	$2 > 1 > 3$	$3 > 1 > 2$
Speculative	$1 > 3 > 2$	$2 > 1 > 3$	$3 > 1 > 2$

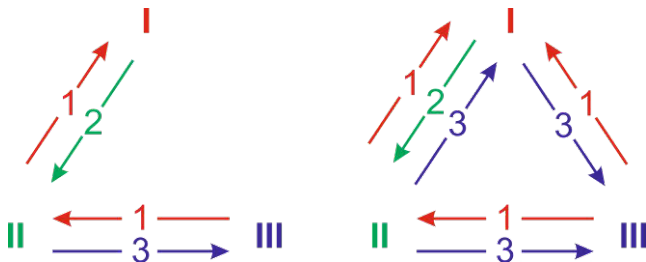


Fig. 2. Trading strategies and resulting trading patterns for the fundamental (left) and speculative (right) equilibria

### Extensions to the Search Model

The original model presented only steady-state equilibria in pure strategies. Subsequent work has considered dynamic and mixed-strategy equilibria (Kehoe, 1993), presenting a more generalised model where agents can alternate their play across their two available trading strategies.

The routes by which a monetary equilibrium could become established have been explored using both analytical and agent-based approaches (Alvarez, 2004). Replicator dynamics have been used to demonstrate analytically the dependence of an ultimate monetary equilibrium on initial conditions such as starting strategies, the storage costs of goods, and the proportions of different agent types in the economy (e.g., Luo, 1999; Sethi, 1999).

The relevance of agent-based approaches to economic modelling is well established (Vriend, 1994; Epstein & Axtell, 1996; Gintis, 1997; Duffy, 2001; Tesfatsion, 2002). Marimon et al. (1990) used classifier systems to allow agents to learn through experience those actions that resulted in positive utility, while Duffy (2001) used experiments with human subjects to appropriately calibrate an agent-based model. Başçı (1999) allowed agents to learn socially through imitation. In general both agent-based and human subject experiments found that social interaction encouraged the use of speculative strategies.

Two hypotheses are tested in the following work: an agent-based replication of the Kiyotaki-Wright model is used to test that Kiyotaki & Wright’s results still hold for small populations; and a numerical simulation of trading strategy evolution is used to test the stability of monetary equilibria in the presence of strategy mutations.

### Finite Population Model

Real economies consist of finite numbers of participants, with interesting economic behaviour exhibited even in very small economies. Agent-based simulation allows the number of interacting agents to be easily selected. The infinite-population model can be approximated with a large population of several thousand agents, or population sizes less than a hundred can be used to see if the same results hold in small communities. Another advantage of running simulations with small populations is that results can be compared to laboratory data from behavioural experiments. Such experiments have typically used less than 30 agents playing a repeated game for less than 100 periods.

### Initialisation

A population size is chosen and an initial population of agents is created, with an equal number of agents of each of the three types. For simplicity the population sizes were chosen to be a multiple of six to ensure an equal distribution across consumption types and to allow all agents to form trading pairs. In the basic model the consumption type also uniquely defines the agent’s trading strategy, with all agents playing fundamental strategies. Agents are initially holding their production goods, representing an economy with no initial endowments or natural resources.

### Trade

Each turn agents are randomly paired into potential trading partnerships and attempt to trade according to their pre-defined trading strategies. This random pairing treats all agents as equally likely to meet, with no memory of past interactions or any attempt to anticipate the outcome of a particular pair. If a successful trade results in an agent holding its consumption good then that agent immediately consumes its holding and gains positive utility by doing so. That agent then immediately produces a new unit of its production good, which becomes its new holding.

At the end of every turn each agent pays the storage cost for its current holding. The utility of an agent consuming its own specific consumption good ( $u$ ) and the storage costs for each good ( $c_1$ ,  $c_2$  and  $c_3$ ) are defined globally and are the

same for each type of agent. Agents record their lifetime utility. When the model is expanded to allow agent trading strategies to evolve, this lifetime utility record will be used as a measure of the fitness of each agent.

### Results of the Agent-Based Model

A single run of the simulation consists of the creation of a population of new agents, the interaction of those agents over a number of turns, and data collection to allow the behaviour of those agents to be summarised.

Data was collected for ease of comparison with results in Kiyotaki & Wright (1989). This consisted of the stocks ( $x$ ) of each good at the end of the turn; the number of transactions ( $t$ ) involving that good during the turn; the 'velocity' ( $v$ ) of each good; and the 'acceptability' ( $a$ ) of each good. These last two values were chosen as two measures of the 'moneyness' of each good, with velocity ( $v = t/x$ ) a more traditional measure (Fisher, 1909) showing the number of transactions weighted by the supply of the good in the economy, while acceptability ( $a = t/o$ ) is the probability that a good will be accepted in trade (Kiyotaki & Wright, 1992), weighting transactions by the number of times a good is offered ( $o$ ).

Results of a single run are shown for a small population of 90 agents (Fig. 3). Solid lines show the levels at the end of each trading turn. Because fundamental equilibrium trading strategies were imposed the system very quickly settles on the equilibrium levels for stocks, transaction, velocities and acceptabilities, taking less than 10 trading turns to do so.

To record these equilibrium levels, averages are calculated for each good from period 10 onwards, and shown as dotted lines. Even for small populations the results are consistent with large- and infinite-population models.

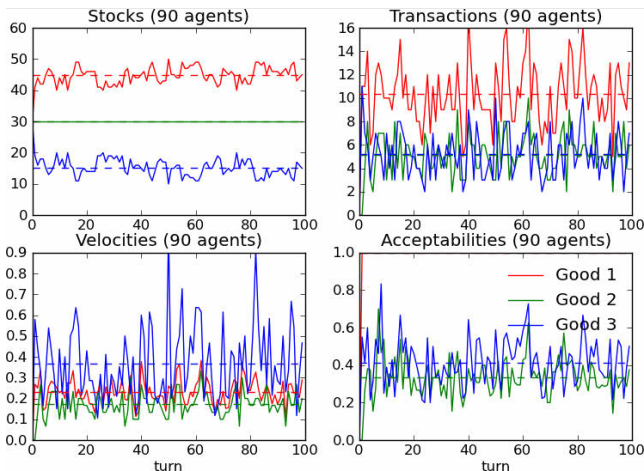


Fig. 3. Results showing stocks, transactions, velocities, and acceptabilities of three goods over time for a single run of the agent-based model with 90 agents

### Evolving Trading Strategies in the Agent-Based Model

The consistency with the infinite population results indicates that an agent-based model is appropriate for studying the emergence of monetary equilibria. The agent-based model can be modified to allow agents to adapt their behaviour. Instead

of imposing unchanging equilibrium trading strategies on the agents, agents are now allowed to adjust their trading strategies based on their relative success. A basic evolutionary algorithm was used to allow successful preferences to be retained, unsuccessful preferences to be replaced, and new or forgotten preferences to emerge.

Instead of imposing equilibrium trading strategies, agents were given trading strategies that were initially completely random. Regardless of consumption type, agents had a 1/6 probability of being assigned one of the initial trading strategies:

$$\begin{array}{lll} 1 > 2 > 3 & 1 > 3 > 2 & 2 > 1 > 3 \\ 2 > 3 > 1 & 3 > 1 > 2 & 3 > 2 > 1 \end{array}$$

These initial trading strategies are unlikely to be beneficial to the agent, as in many instances they will lead to an agent rejecting its consumption good. However, the evolutionary model will allow agents to adapt their trading strategies to match those that have been successful in the previous generation, allowing us to test whether this model is sufficient for a monetary equilibrium to emerge.

### Trading Strategy Fitness

The simulation is broken down into a number of generations ( $G$ ), each consisting of a number of trading periods ( $T$ ). At the start of the first generation agents are given random trading strategies as described above. Play within a single generation is the same as described above, with agents being randomly paired and trading when both agents in a pair prefer their partner's holding to their own, given their current trading strategy. Agents keep track of their lifetime utility, which increases whenever they receive and consume their consumption good, and decreases by the storage cost of their holding at the end of each trading turn.

At the end of each generation agents are ranked by their total utility across all the trading turns in that generation. Agents who consume a relatively large amount, or spend fewer turns carrying the goods with the highest storage costs, will have the highest utilities within that generation. This total utility level is used as a measure of the fitness of that agent's trading strategy, with higher fitness trading strategies more likely to survive into subsequent generations.

### Imitation and Mutation of Trading Strategies

The agent population is first divided by consumption type. Within each consumption type, the 80% least successful agents are discarded. Each of the most successful 20% of agents then produces four offspring, so that the population size remains unchanged between generations.

Offspring are initially a perfect copy of their parent, with the same consumption type and the same trading strategy. There is then a 10% chance that each child will slightly mutate its trading strategy by swapping the order of two goods in its priority list. The two goods that are swapped in this way are chosen randomly with equal probability of any two goods being selected, i.e. the first and second item may be swapped with  $p = 1/20$ , or the second and third item may be swapped with  $p = 1/20$ .

This mutation mechanism means that at most two items are swapped in the trading strategy, with zero chance of a larger mutation or multiple mutations within a generation.



## Results of the Evolutionary Model

In all cases a population size of 300 agents (100 of each consumption type) is chosen. Generations consist of  $T = 1000$  trading turns, and trading strategies are reproduced and mutated across  $G = 50$  generations.

Results of the evolutionary model are plotted as charts that show the dominant trading strategies of each type of agent against generational time. Coloured squares are used to show the proportion of each type of agent using a given trading strategy in a given generation. Colours represent agent consumption type (type I in red, type II in green and type III in blue), with the intensity of that colour showing the proportion of the population who are choosing that trading strategy. Saturated colours represent a trading strategy chosen by all or most agents of a particular type, while very weak colours signify a trading strategy chosen by few or no agents of that type.

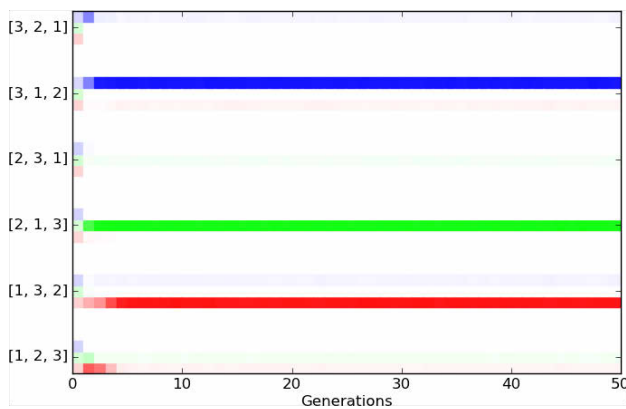


Fig. 4. Agent-based evolution of speculative strategy ( $u = 100$ ,  $c_1 = 1$ ,  $c_2 = 4$ ,  $c_3 = 9$ , average over 20 runs)

When utility of consumption is suitably high, type I agents benefit by adopting the speculative trading strategy, as shown in Fig. 4. Although it takes several generations for agents to adapt, they ultimately settle on the speculative equilibrium.

With lower utilities for consumption ( $u < 20$ ), holding costs become increasingly significant and small populations of agents may not discover the fundamental equilibrium.

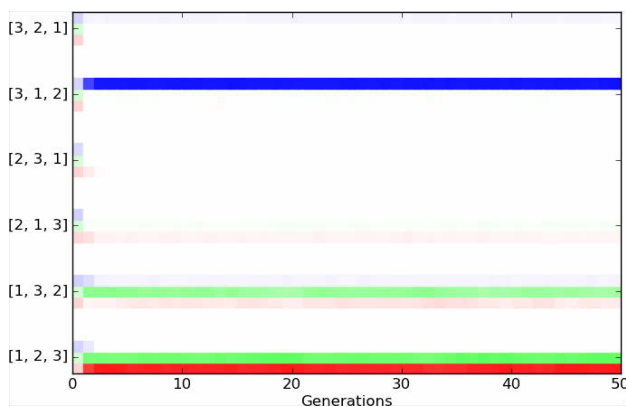


Fig. 5. Failure to discover fundamental strategy ( $u = 10$ ,  $c_1 = 1$ ,  $c_2 = 4$ ,  $c_3 = 9$ , average over 20 runs)

Fig. 5 shows that type II agents prefer a trading strategy that permanently minimises their storage costs by accepting the lowest cost good (1) and never releasing it. Surprisingly, they do not learn to accept their own consumption good. This shows that agents in small-population runs of our model can adopt and persist with trading strategies that were not predicted by analytic approaches. The finding is intriguing but we expect that it would not occur given a larger population and indeed our focus here is not on the discovery of consumption goods (surely a no-brainer in evolutionary terms) but on the origins of monetary exchange.

## Evolving Trading Strategies in a Large Population Model

An alternative to simulating the evolution of individual agent behaviours is to simulate the evolution of the population as a whole. In their original paper, Kiyotaki & Wright computed elements  $p_{ij}$  of a population array  $\mathbf{p}$ , with the elements corresponding to the proportion of type  $i$  agents holding good  $j$  in the steady state reached after a number of trading steps. In Kiyotaki & Wright these 6 elements of  $\mathbf{p}$  were sufficient to completely describe the population because each consumption type held a fixed (fundamental or speculative) trading strategy.

To study the evolution of trading strategies using a population array, new elements need to be added to allow agents of the same consumption type to use different trading strategies. Following a methodology similar to Luo (1999) and Sethi (1999) who studied this problem analytically, the population array  $\mathbf{p}$  is reconstructed using 12 elements, with each element of the array representing a triplet of consumption type, holding and trading strategy.

Each consumption type is now permitted two trading strategies, both of which still prioritise that type's consumption good. The interesting question in monetary search is not whether an agent discovers his consumption good (which he must in order to gain any utility), but how that agent treats non-consumption goods in a monetary capacity. In this three-good system the agent can prioritise the remaining two non-consumption goods in two ways: either it can prefer to hold its cheaper non-consumption good (a fundamental trading strategy) or it can prefer to hold its more expensive non-consumption good (a speculative trading strategy).

This is a slightly different labelling than employed in Sethi (1999), which treats consumption types as preferring to hold their production good or their non-production good. In the case of type I and type III agents, their production good is also the cheaper of their two non-consumption goods. But in the case of type II agents, their production good is the more expensive of their non-consumption goods.

## Initialising the population

The population array is composed of twelve elements corresponding to one of three consumption types, each of whom may hold one of their two non-consumption goods and one of two trading strategies (fundamental or speculative). Each consumption type comprises one third of the population,

and the population is initialised so that all agents are holding their production good. The proportion of each trading type following each of their two possible trading rules can be varied. As an example, if all consumption types started with equal proportions playing each possible trading strategy, the initial elements of the population array would be:

$$\begin{array}{lll} p_{I(2)123} = 1/6 & p_{II(1)213} = 0 & p_{III(1)312} = 1/6 \\ p_{I(3)123} = 0 & p_{II(3)213} = 1/6 & p_{III(2)312} = 0 \\ p_{I(2)132} = 1/6 & p_{II(1)231} = 0 & p_{III(1)321} = 1/6 \\ p_{I(3)132} = 0 & p_{II(3)231} = 1/6 & p_{III(2)321} = 0 \end{array}$$

where the subscripts represent consumption type-(holding)-trading strategy.

### Trading to a steady state in holdings

After initialisation the simulation iterates through an outer loop. At the beginning of the iteration all agents' holdings are reallocated to the production goods of that type. Agents already have trading strategies, either from a previous iteration or from initialisation.

The population shares are updated to reflect repeated matching by agents for the given trading strategies. Any particular match between type  $i$  and  $j$  will occur with probability  $p_i p_j$ , with trade occurring if it is beneficial to both members of the pair as in all earlier models.

If as the result of a match an agent ends up holding its consumption good, it immediately consumes it and replaces it with its production good. The population share resulting from such a match is therefore added to the element corresponding to that agent's consumption type, trading strategy and production good. After multiple trading steps a steady-state in goods is reached.

### Replication of successful trading strategies

When the holdings reach a steady state – i.e. the holdings on two subsequent turns are sufficiently similar (within a specified tolerance level, set as  $10^{-6}$  for the results in this paper) – the trading phase of the simulation terminates, and the time-discounted expected lifetime utilities of different types of agent are calculated.

Agents are given a great degree of foresight while calculating these expected lifetime utilities. For the reported results, agents were allowed to look-ahead 100 periods using a discount factor of  $\beta = 0.9$ . Any calculation with more than about 50 periods is a good approximation to an infinitely-lived, perfectly rational agent.

Agents of a given consumption type and holding are then allowed to imitate each other's strategies based on their relative expected utilities. A discretised version of the replicator equation (Weibull, 1995) is used to adjust the population shares across trading strategies for each consumption type-holding pair. The population share for a given consumption type ( $t$ ), holding ( $h$ ) and trading strategy ( $s$ ) is updated as:

$$p_{t(h)s} = p_{t(h)s} \left( 1 + r \left( U_{t(h)s} - \frac{p_{t(h)s} U_{t(h)s} + p_{t(h)s'} U_{t(h)s'}}{p_{t(h)s} + p_{t(h)s'}} \right) \right)$$

where  $s'$  is the alternative trading strategy for the same consumption type-holding pair,  $U_{t(h)s}$  are the expected discounted lifetime utilities already calculated, and  $r$  is a scaling factor used to represent selection pressure. The proportion of the population using a particular trading strategy increases if that strategy yields a higher expected utility than the population-weighted average of the two alternative strategies, at a speed proportional to the difference. The proportion playing the less successful strategy will shrink.

After performing a single trading strategy update step, the population is reinitialised to hold their production goods and the next iteration begins with a new round of trading to a steady state in holdings. This process continues until successive updates of the entire outer loop produce no further change in trading strategy share.

After each trading step and trading strategy update the population is re-normalised to ensure that small numerical errors do not result in the creation or destruction of holdings (during the trading steps) or a re-allocation across consumption types (during strategy updates).

The same general results are obtained for less far-sighted agents. The number of future time periods considered in expected utility calculations was chosen to allow relatively rapid convergence to a trading strategy equilibrium, but limiting this amount of foresight only slows the learning process, it does not change the end result. As long as agents consider at least one period into the future, they are able to appreciate the benefits of a monetary good as a medium of exchange.

## Results of the Large Population Model

A variety of setups were used to explore conditions under which the different equilibria of the Kiyotaki-Wright model could be reached.

Results are visualised in the trading strategy space of the three agent types. Each consumption type can play one of two strategies: either the fundamental trading strategy that favours holding the lower numbered good (the cheaper good in the conventional setup of  $c_1 < c_2 < c_3$ ), or the alternative speculative trading strategy that favours the higher numbered (more costly) good.

After each trading strategy update, the proportion of each consumption type playing that type's fundamental strategy was recorded, and these proportions used to label the axes of a cube that describes the strategies of all agents in the economy. The 0 of the axis corresponds to all agents playing their speculative trading strategy, while 1 corresponds to all agents playing their fundamental trading strategy.

A selection of starting points was chosen (27 points formed by all possible combinations of [0.25, 0.5 and 0.75] across the three consumption types) and trading strategies allowed to evolve under the rules described above. When plotted these evolving trading strategies tend to trace paths from a uniform three dimensional grid in the centre of the trading strategy space towards one of the equilibria at the corners of the cube. This equilibrium was dependent only on utilities and costs, and not on the particular starting trading strategies.

Results show a representative evolution of trading strategies for the starting mix of trading strategies (1/2, 1/2, 1/2).

Storage costs and utilities have been chosen for consistency with Duffy (2001), with  $c_1 = 1$ ,  $c_2 = 4$  and  $c_3 = 9$ .

### Fundamental Equilibrium

With  $u = 10$  (Fig. 6), all paths rapidly converge on the fundamental equilibrium (1,1,1), with all agents preferring lower storage cost goods to higher storage cost goods (type I:  $1 > 2 > 3$ , type II:  $2 > 1 > 3$ , type III:  $3 > 1 > 2$ ).

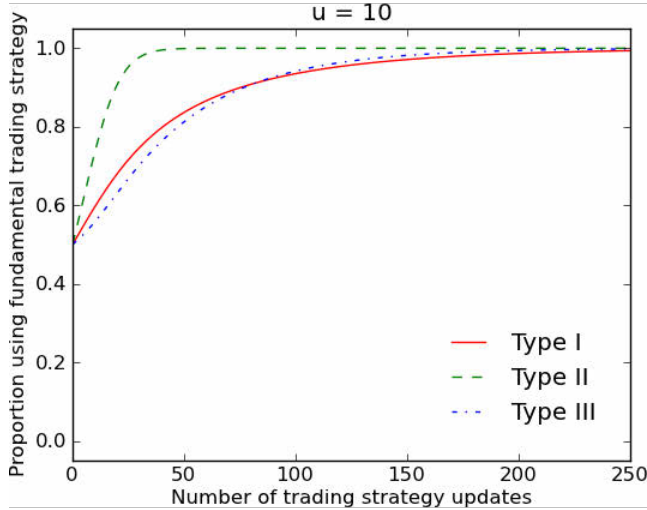


Fig. 6. Fundamental equilibrium (rapid convergence)

Increasing the utility of consumption increases the incentive for type I agents to speculate and experiment with holding a monetary good, as the benefits of more frequent trade are greater relative to the fixed costs of holding goods. With  $u = 20$  and costs unchanged (Fig. 7) trading strategies still converge on the fundamental equilibrium, but far more slowly.

Although type II agents still converge rapidly on their fundamental trading strategy, there is very little evolutionary pressure for a mixed population of both fundamental and speculative type I agents to move towards the fundamental trading strategy, as the expected utilities of either trading strategy are very similar.

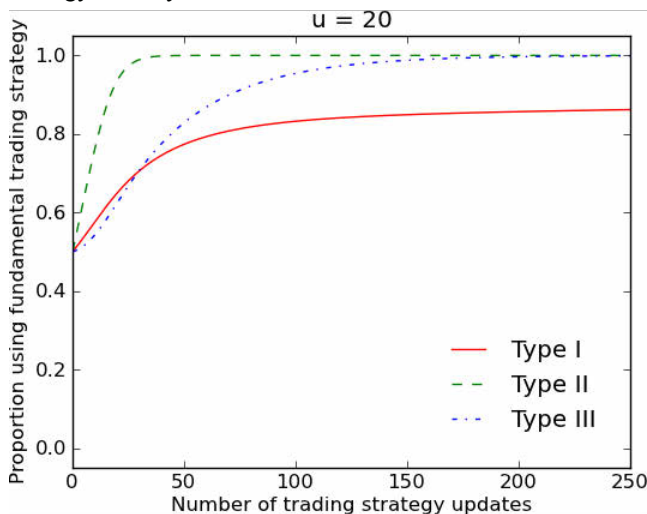


Fig. 7. Fundamental equilibrium (slow convergence)

### Speculative Equilibrium

When the utility of consumption is sufficiently high, type I agents can increase their expected utility by accepting good 3 from type II agents. The additional expense of holding this high storage cost good is offset by the increased expectation of direct trade with type III agents for good 1, the type I agents' consumption good.

Kiyotaki & Wright calculate the critical level at which type I agents should speculate as dependent on the level of utility, the holding costs and the proportions of type II and type III agents who are holding good 1. Type I agents should speculate if  $c_{13} - c_{12} < (p_{31} - p_{21})\beta u_1$  (Kiyotaki & Wright, 1989).

This can be seen when the utility of consumption is set sufficiently high, with  $u = 100$  in Fig. 8. Convergence to this speculative equilibrium occurs rapidly, with all three consumption types converging on their equilibrium strategy in similar timescales.

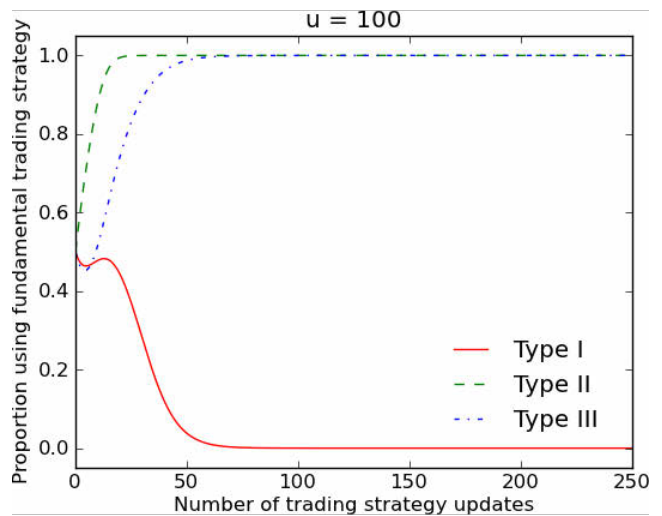


Fig. 8. Speculative equilibrium

To produce Fig. 8 an appropriate speed of trading strategy replication ( $r$ ) needed to be chosen. Lower values of  $r$  mean that the system takes longer to reach an equilibrium, but the more gradual replication of trading strategies stops strategies from becoming caught at alternative equilibria.

This occurs because a trading strategy can only be replicated if it still exists within the population. Once entirely eliminated, the replicator equation used above will not allow a trading strategy to re-emerge, as it has a zero weighting in the population average. If  $r$  is too high those paths that pass close to the corners (1,1,0) and (1,1,1) may become trapped at these alternative equilibria before reaching the speculative equilibrium (0,1,1).

As well as slowing down the speed of convergence, another way to ensure that the system does not approach a sub-optimal equilibrium due to these numerical errors is to introduce a small mutation rate that allows extinct trading strategies to reappear. In the cases discussed above such a mutation rate will only temporarily move the system away from the equilibrium, but becomes interesting in the case of a mixed equilibria system.

## Multiple Equilibria

This framework which had been used to reproduce the evolutionary dynamics described analytically in Sethi (1999) can also be used to consider the case of identical goods, proposed in an example in Luo (1999). If goods are either identical or very similar, the particular type of money used in the economy may have a sensitive dependence on the initial mix of agent trading strategies.

By relaxing the cost ordering of Kiyotaki-Wright and setting the storage cost of goods equal ( $c_1 = c_2 = c_3 = 0$ ), the particular monetary equilibrium depends only on the initial trading strategies used by agents, as shown in Fig. 9 where  $(1/3, 2/3, 1/3)$  is a critical point around which trading strategies significantly diverge. Trading strategies started at the 125 points formed by allowing each starting trading strategy to diverge from this critical point by  $[-0.02, -0.01, 0, 0.01, 0.02]$ .

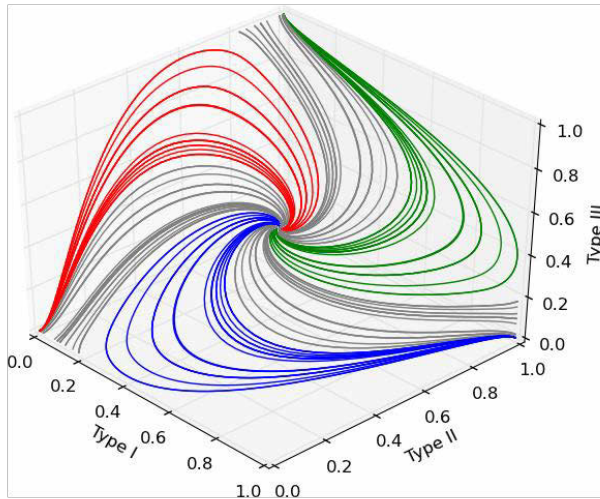


Fig. 9. Multiple equilibria for identical goods ( $u = 100, c_1 = c_2 = c_3 = 0$ )

Trading strategies can end at one of the three 'speculative' equilibria, shown in red  $(0,0,0)$ , green  $(0,1,1)$  and blue  $(1,1,0)$ , or at an equilibrium (shown in grey) where one consumption type continues to be composed of agents playing both 'fundamental' and 'speculative' strategies.

The labels 'fundamental' and 'speculative' are no longer meaningful as all goods have an identical holding cost. However, the particular good that is used as a medium of exchange depends on the equilibrium point that is reached, which depends only on the starting mix of trading strategies:

- $(0,1,1)$ : Type I accept good 3, type II accept good 1
- $(0,0,0)$ : Type III accept good 2, type I accept good 3
- $(1,1,0)$ : Type II accept good 1, type III accept good 2

## Strategy Mutation

With trading strategy imitation described by the replicator equation, once agents reach an equilibrium (at a corner or edge of the trading strategy space) they will remain there, as there is no process for forgotten strategies to be rediscovered.

Allowing a small degree of trading strategy mutation after the replication phase allows forgotten trading strategies to return. Each consumption type-holding pair is mutated

independently. In each case a random number is drawn uniformly from the interval  $[-0.001, +0.001]$  and multiplied by the proportion of the population playing either trading strategy in this pair. The proportion of agents playing the fundamental trading strategy is then increased by this amount, and the proportion playing the speculative trading strategy decreased by the same amount, with a normalisation step to ensure that this does not result in either proportion becoming less than zero.

In the case of the fundamental and speculative equilibria discussed above these mutations have little effect. Mutations cause the trading strategies to fluctuate around the equilibrium, but strategy replication takes the system back towards it.

However, in the case of the mixed equilibria, strategy mutations can drive the system around the edges of the trading strategy space, permitting sudden transitions from one monetary equilibrium to another. Starting from the equilibrium at  $(0,1,1)$ , the results of strategy mutations are shown in Fig. 10.

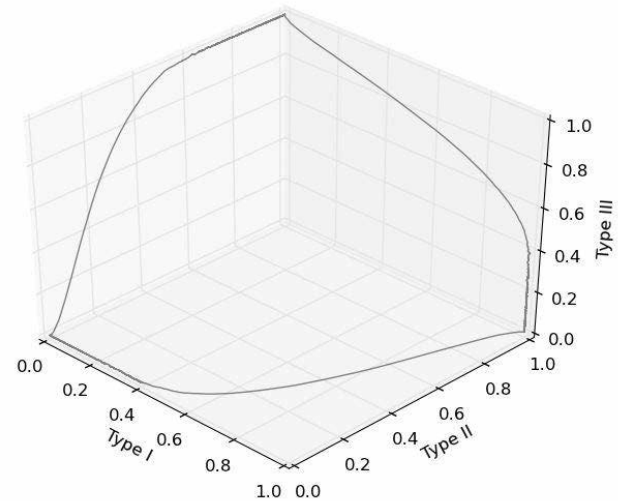


Fig. 10. Trading strategy mutation shifts monetary equilibria

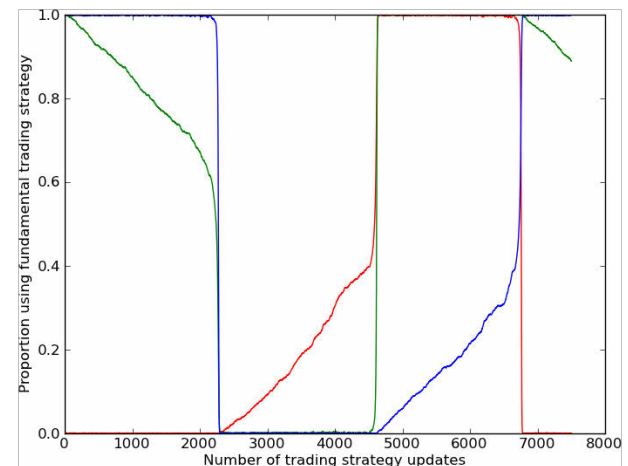


Fig. 11. Cycling through trading strategy equilibria



Fig. 11 shows the speed of these transitions. Trading strategies initially move slowly along the edge of the trading strategy space. Along these edges one consumption type is split into players playing both fundamental and speculative trading strategies, while the other two consumption types are playing a single strategy. The mutation step allows these single-strategy players to experiment with their alternative strategy, which provides an additional incentive for the two-strategy player to shift in favour of the second strategy. Ultimately a critical point is reached where enough of the current two-strategy players are playing the second strategy to cause the rapid transition to a new equilibrium. At this point the two-strategy player has an incentive to play only their second strategy, and a new consumption type begins experimenting with their alternative strategy.

This shows how the gradual shift in the acceptance of one type of money by one type of agent can tip the population structure to the point where an entirely new good becomes accepted as money.

For instance, in the initial move from (0,1,1) towards (0,0,1), with a sudden transition to (0,0,0): Initially good 3 is used as money by type I and good 1 is used as money by type II. Type II agents then increasingly refuse to accept good 1, driving the system towards an equilibrium where their own production good is the unique monetary good. However, at a critical point the system shifts as type III agents begin accepting good 2 as a monetary good.

This process repeats. In each case an agent shifts its trading strategies in favour of creating a monopoly in money production, ultimately resulting in a shift that begins a cycle where the original agents production good is rejected as money entirely.

## Discussion

It is encouraging that the models of Kiyotaki & Wright (1989) and Sethi (1999) can be reproduced in both agent-based and numerical simulations that support the original analytic results. The findings presented here explicitly confirm the infinite-population based estimates of Sethi in the limit of very small population size and in the presence of noise in the evolutionary dynamics.

Speculative, fundamental, and mixed equilibria can each be supported if appropriate consumption utilities and storage costs are chosen. If goods are homogenised by setting their storage costs to be equal, multiple equilibria can also be supported; this finding provides a way in to modelling problematic phenomena such as competing currencies (Hayek, 1976) or monetary collapse.

There are many ways in which this framework could be extended. One of the most obvious would be to consider more realistic economies in which more than three types of agents trade more than three types of goods; in which goods differ in their properties such as durability; and in which prices can vary. Another line of extension would be to investigate the effect an evolution that is constrained to real-world social networks has on monetary search. The current model assumes a complete trading network, where any two agents may meet and attempt to trade with equal probability. Real trading environments tend to have strong cultural, social or geographical roots, suggesting that investigating the

evolution subject to more constrained interaction patterns could be an important step in motivating the maintenance of multiple competing currencies.

## Acknowledgments

This work was supported by an EPSRC Doctoral Training Centre grant (EP/G03690X/1).

## References

- Alvarez, A. (2004), Learning to choose a commodity-money: Carl Menger's theory of imitation and the search monetary framework, *The European Journal of the History of Economic Thought*, 11:53-78
- Başçı, E. (1999), Learning by imitation, *Journal of Economic Dynamics & Control*, 23:1569-1585
- Beinhocker, E. D. (2007), *The Origin of Wealth: Evolution, Complexity and the Radical Remaking of Economics*, Random House Business Books, London
- Duffy, J. and Ochs, J. (1999), Emergence of Money as a Medium of Exchange: An Experimental Study, *The American Economic Review*, 89:847-877
- Duffy, J. (2001), Learning to Speculate: Experiments with Artificial and Real Agents, *Journal of Economic Dynamics and Control*, 25:295-319
- Epstein, J. M. and Axtell, R. (1996), *Growing Artificial Societies: Social Science from the Bottom Up*, The Brookings Institution, Washington D.C.
- Fisher, I. (1909), A Practical Method of Estimating the Velocity of Circulation of Money, *Journal of the Royal Statistical Society*, 72(3):604-618
- Gintis, H. (1997), A Markov Model of Production, Trade and Money: Theory and Artificial Life Simulation, *Computational & Mathematical Organization Theory*, 3:1:19-41
- Hayek, F. A. (1976), *Denationalisation of Money: The Argument Refined*, Institute of Economic Affairs, London
- Jevons, W. S. (1875), *Money and the mechanism of exchange*, D. Appleton and Company, New York
- Kehoe, T. J., Kiyotaki, N. and Wright, R. (1993), More on Money as a Medium of Exchange, *Economic Theory*, 3:297-314
- Kiyotaki, N. and Wright, R. (1989), On Money as a Medium of Exchange, *The Journal of Political Economy*, 97:927-954
- Kiyotaki, N. and Wright, R. (1992), Acceptability, Means of Payment, and Media of Exchange, *Federal Reserve Bank of Minneapolis Quarterly Review*, 16(3):18-20
- Luo, G. Y. (1999), The evolution of money as a medium of exchange, *Journal of Economic Dynamics and Control*, 23:415-458
- Marimon, R., McGrattan, E. and Sargent, T. J. (1990), Money as a Medium of Exchange in an Economy with Artificially Intelligent Agents, *Journal of Economic Dynamics and Control*, 14:329-373
- Rodvalho, W. M., Vinhal, C. D. N. and da Cruz Jr., G. (2010), Studying the Emergence of Money by Means of Swarm Multi-agent Simulation, *Advances in Artificial Intelligence – IBERAMIA 2010, Lecture Notes in Computer Science*, 6433:296-305
- Sethi, R. (1999), Evolutionary stability and media of exchange, *Journal of Economic Behavior & Organization*, 40:233-254
- Tesfatsion, L. (2002), Agent-Based Computational Economics: Growing Economies from the Bottom Up, *Artificial Life*, 8(1):55-82
- Vriend, N. J. (1994), Artificial Intelligence and Economic Theory, *Many-Agent Simulation and Artificial Life* (1994):31-47
- Weibull, J. W. (1995), *Evolutionary Game Theory*, The MIT Press, Cambridge, MA

# BEYOND THE SCHELLING'S SEGREGATION MODEL: IS IT EQUIVALENT TO BE REPULSED BY DISSIMILAR RATHER TO BE ATTRACTED BY SIMILAR?

Philippe Collard

University Nice Sophia-Antipolis

I3S laboratory

France

philippe.collard@unice.fr

## Abstract

From the Schelling model of segregation, we derive models of group formation that shed light on segregation or mixing patterns observed in spatial grid networks. Individuals have types and see type-dependent benefits or drawback from their neighbours: this leads each one to be attracted or repulsed by its own like or unlike. This framework allows to studying many spatial phenomena that involve individuals making location choices as a function of the characteristics and choices of their neighbours. Our goal is to grow social structures *in silico* and to ask if related micro-specifications generate similar macro-phenomena of interest. Regarding (i) the amount of segregation-mixing, (ii) the congruence between micro-motives and macro-behaviour and (iii) the nature of frontier between clusters, we examine the properties of the steady-state equilibrium.

## Introduction

This article is based on the Schelling's checkerboard model of residential segregation ?; this model has become one of the most cited and studied models in many domains as economics, sociology, complex systems science... ? ? ?. In the Schelling model the world is a 2-dimensional *grid* composed either by locations occupied by agents or by vacant ones. The perception of an agent is centred on its local neighbourhood only. There are two types of agents. The behaviour of one agent consists to move away on if it is not satisfied. Satisfaction is relative to the *proportion* of agents with a dissimilar type in the neighbourhood.

In this paper, the goal is to grow social structures *in silico* and to ask if related micro-specifications generate similar macro-phenomena of interest. We define a generic framework in order to shape several models with various micro-motives. This family of models allows us to show that a wide variety of macro-behaviours can emerge despite the fact that they come from a same mould and share some common features. In particular, we propose to consider the following issues: Q1) Is it equivalent to flee regarding the proportion of dissimilar agents among the neighbourhood agents rather than the number of dissimilar agents in the full set of neighbours? Q2) Is it equivalent to be repulsed (resp. attracted) by

dissimilar neighbours rather to be attracted (resp. repulsed) by similar neighbours ?

## The AR models

To answer these questions, we define the family of attraction-repulsion models (AR models). For all these models, the world is a 2-dimensional *grid* where nodes-cells represent either locations occupied by agents or vacant ones. A vacant cell could be occupied by an agent later. The perception of an agent is centred on its local neighbourhood only. One assumes that the neighbourhood of an agent is constituted both by the nearest agents and the vacant cells surrounding him: in this paper we consider the *Moore* neighbourhood composed of the eight nearest cells surrounding it. Let be  $V$  the set of vacancy cells and  $A$  the set of agents. We assume that the number of agents ( $\#A$ ) is conserved and the total volume in which they move ( $\#A + \#V$ ) is constant. The density of agents is the ratio  $d = \frac{\#A}{\#A + \#V}$ . There are two types of agents and each agent has its own type. The agent's type can never change. For convenience we will denote by a color, *blue* and *yellow*, the two possible types. Yellow and blue agents can be interpreted as individuals representing any two groups in society (two genders, smokers and non-smokers, etc.) Let  $B$  (resp.  $Y$ ) the set of agents in the *blue* type (resp. *yellow* type). So,  $\#B + \#Y = \#A$  and, at the global level, the basic hypothesis is ( $\#B = \#Y$ ). Each agent is satisfied or unsatisfied according to its own type and the type of its neighbours.

## Micro motives

In the Schelling model one agent moves if its utility  $u$  falls below a certain threshold ( $u \leq \tau$ ). In this section we generalize this model in considering different utility functions and we propose to study the two cases where the utility is either below or above the threshold  $\tau$ .

**Utility functions** For each agent  $a_i$ , considering only the agents in its neighbourhood, we define the two utility functions:

$$\Delta_i = \begin{cases} \frac{\delta_i}{\sigma_i + \delta_i} & \text{if } \sigma_i + \delta_i \neq 0 \\ 1 & \text{else} \end{cases} \quad (1)$$

$$\Sigma_i = \begin{cases} \frac{\sigma_i}{\sigma_i + \delta_i} & \text{if } \sigma_i + \delta_i \neq 0 \\ 1 & \text{else} \end{cases} \quad (2)$$

where  $\delta_i$  (resp.  $\sigma_i$ ) is the number of dissimilar (resp. similar) nearby for the agent  $a_i$ .

In the same way, considering the set of all the cells in the neighbourhood of the agent, we define the two utility functions:

$$\Delta'_i = \frac{\delta_i}{neighbourhoodSize} \quad (3)$$

$$\Sigma'_i = \frac{\sigma_i}{neighbourhoodSize} \quad (4)$$

Let's note that  $\Delta_i + \Sigma_i = 1$ , while, in general,  $\Delta'_i + \Sigma'_i \neq 1$ .

So, given a threshold  $\tau$  in the range  $[0, 1]$ , there are potentially eight ways to express a condition to be satisfied (Table ??). Let's note that the model  $M_0$  corresponds to the Schelling model of segregation.

Table 1: Potential conditions to be satisfied

Model	$u \leq \tau$	Model	$\tau \leq u$
$M_0$	$\Delta \leq \tau$	$M_4$	$\tau \leq \Delta$
$M_1$	$\Delta' \leq \tau$	$M_5$	$\tau \leq \Delta'$
$M_2$	$\Sigma \leq \tau$	$M_6$	$\tau \leq \Sigma$
$M_3$	$\Sigma' \leq \tau$	$M_7$	$\tau \leq \Sigma'$

**Complement model** Two models  $M_i$  and  $M_j$  are said to be *complementary* iff the conditions to be satisfied are logical complementary; then, we note  $M_j = \overline{M_i}$ .

Obviously the equation  $\overline{\overline{M}} = M$  holds. We have  $M_4 = \overline{M_0}$ ,  $M_5 = \overline{M_1}$ ,  $M_6 = \overline{M_2}$  and  $M_7 = \overline{M_3}$ .

**Dual model** Two models  $M_i$  and  $M_j$  are said *dual* iff the conditions to be satisfied are dual; we note  $M_j = M_i^*$ . We say that two conditions  $(U_i \leq \tau_i)$  and  $(\tau_j \leq U_j)$  are *dual* iff  $(U_i, U_j) \in \{(\Delta, \Sigma), (\Sigma, \Delta), (\Delta', \Sigma'), (\Sigma', \Delta')\}$ .

Obviously the equations  $\forall i, M_i^{**} = M_i$  and  $\overline{(M_i^*)} = (\overline{M_i})^*$  hold. More, as  $\Delta + \Sigma = 1$ ,  $M_0^* = M_0$  and  $M_2^* = M_2$ . Finally, we have  $M_4 = M_2^* = M_2$ ,  $M_5 = M_3^*$ ,  $M_6 = M_0^* = M_0$  and  $M_7 = M_1^*$ .

**AR models** All this leads us to consider six models of segregation/mixing only. For each model, table ?? gives the local condition under which an agent is satisfied. Let's note that all the AR models are defined from the two models  $M_0$  and  $M_1$  only.

Table 2: AR models: condition to be satisfied

$M_0 \equiv (\Delta \leq \tau)$	$M_1 \equiv (\Delta' \leq \tau)$	$M_1^* \equiv (\tau \leq \Sigma')$
$\overline{M_0} \equiv (\tau \leq \Delta)$	$\overline{M_1} \equiv (\tau \leq \Delta')$	$\overline{M_1^*} \equiv (\Sigma' \leq \tau)$

## Rules governing agents movement

In the standard Schelling's models, as soon as an agent is unsatisfied, it moves to another place where it becomes satisfied. So, the local behaviour is an optimisation process which needs any agent to access information about any cell-location in order to compute its utility function in the target cell.

To stay is in the spirit of the complex systems paradigm, in AR models there is no decision-rule to decide whether or not an agent gains an advantage by means of migration towards a target cell. To find a new place, an unsatisfied agent uses a simple rule (what we call the *Eulogy to Fleeing (EF)* rule): *a location is randomly chosen from the world and the agent moves into it if and only if this location is vacant*. Consequently, an agent may move at random towards new locations by allowing utility-increasing or utility-decreasing moves. As the moves do not equate to immediate benefits, it is challenging to predict the overall emerging effect.

## Macro behaviour

For the six AR models, simulations will show that, applying the *EF rule*, in many cases, the population reaches a state where all the agents are satisfied or, at least, a large proportion of agents are satisfied. In the first case we will say that there is *convergence* and in the second case *quasi-convergence*. In this paper, we do not discuss the conditions which guarantee convergence towards equilibrium or quasi-equilibrium; we select system conditions in which one of these two cases occurs. Let's note that the *EF* rule has already been used within Schelling's models, leading the system towards equilibrium ?.

**Segregation vs. mixing** According to its micro-motive, we may assume that each AR model leads at the global level of the population either towards *segregation* or *mixing*. For a given model, regarding if the micro-motive is based on *repulsion* or *attraction* in the face to *similar* or *dissimilar* neighbours, we indicate if the macro-behaviour should be rather either segregation or mixing (Table ??). Given to models  $M_i$  and  $M_j$ :

- if  $M_j$  and  $M_i$  are *complementary*, then the respective macro-behaviour and micro-motives are opposed but the target neighbours are identical.
- if  $M_j$  and  $M_i$  are *dual*, then the respective macro-behaviour are identical but the micro-motives and target neighbours are opposed.

Table 3: AR models: attraction vs. repulsion

Model	Micro motive	Target neighbours	Macro behaviour
$M_0$	repulsion	dissimilar	segregation
	attraction	similar	segregation
$M_1$	repulsion	dissimilar	segregation
$M_1^*$	attraction	similar	segregation
$\overline{M}_0$	attraction	dissimilar	mixing
	repulsion	similar	mixing
$\overline{M}_1$	attraction	dissimilar	mixing
$\overline{M}_1^*$	repulsion	similar	mixing

Because the  $M_0$  (resp.  $\overline{M}_0$ ) model is *self-dual*, it can be view either as a model of repulsion against its dissimilar (resp. similar) agent-neighbours or a model of attraction for its similar (resp. dissimilar) agent-neighbours; in both case there is emergence of segregation (resp. mixing).

**An index to measure the segregate-mixing ratio** In order to gain some insight into the segregation-mixing level, it is necessary to measure the global state of segregation of the world. We reformulate measures proposed by ?, ? and ?.

First, for each time  $t$ , we define a global measure of *similarity* as:

$$s(t) = \frac{1}{\#A} \sum_i^{\#A} (1 - \Delta_i(t))$$

Then we define the *segregate-mixing* index by:

$$segMix = \begin{cases} \frac{s - s_{rand}}{1 - s_{rand}} & \text{if } s \geq s_{rand} \\ \frac{s - s_{rand}}{s_{rand}} & \text{else} \end{cases} \quad (5)$$

where  $s_{rand}$  is the expected value of the measure  $s$  implied by a random allocation of the agents in the world. So, a zero *segMix* index corresponds to a random positioning of the agents. The maximum value of 1 corresponds to a configuration with two homogeneous patterns (complete segregation into two same-colour groups), whereas negative values point towards highly mixed populations.

**An index to measure the congruence between micro-motives and macro behaviour** We define the *micro-Macro* index as :

$$mM = \begin{cases} \frac{\overline{u}_i}{\tau} & \text{if } \tau \geq \overline{u}_i \\ \frac{1 - \overline{u}_i}{1 - \tau} & \text{else} \end{cases} \quad (6)$$

where  $u$  is the utility function and  $\overline{u}_i = \text{mean}_i(u_i)$ . The value 1 corresponds to the theoretical optimal case where dynamics build exactly the needed liveable configurations: in this case the congruence between micro and macro levels is maximum. The value 0 corresponds to the extreme

case where dynamics build much more liveable configurations than necessary to reach the emergent macro-state: in this case there is no congruence between micro and macro levels.

**Frontier** A frontier is a generic concept that has different instantiations depending on the context in which it is considered. A common class of frontiers is found in the geographical domain, where they appear as fronts. The interest we take in this concept lies on two aspects: from a *static* standpoint, a frontier enables the separation of incompatibilities; however, as it allows at the same time some form of communication between them, a *dynamic* perspective is also relevant. So, we consider a frontier as a structure which both determines the “borderland” between two aggregates of opposite types and allows communication between them ?.

**Definition** A *frontier* is composed of the cell-locations where contact occurs between two dissimilar agents. We consider contacts as being of two types: *direct* or *indirect*. A direct-contact refers to agents being directly linked, whereas an indirect-contact is mediated through a vacant location. In the real world, a direct-contact can be exemplified through the contact of a healthy person with a person having a communicable disease, whereas an indirect-contact is achieved through some intervening medium *e.g.* air. Let  $D$  (resp.  $I$ ) the set of direct (resp. indirect) contacts:

$$D = \{(a_i, a_j) \in B \times Y \mid a_j \in N(a_i)\}$$

$$I = \{(a_i, a_j, v) \in (B \times Y \times V) \mid v \in N(a_i) \cap N(a_j)\}$$

where  $N(a_k)$  is the neighbourhood of the agent  $a_k$ . Then we define the frontier as:  $F = (A_F \cup V_F, E_F)$ ; where  $A_F$  is the subset of agent-cells that are at least one coordinate of an element of  $D$  or  $I$ ,  $V_F$  is the subset of vacant cells that are at least one coordinate of an element of  $I$  and  $E_F$  is the set of links between neighbouring cells of  $F$ .

**Characteristics of a Frontier** To enable consistent comparisons of frontiers, we take into account their importance relative to the entire world and their openness. Thus, we define what we mean by a frontier’s *occupancy* and *porosity*. These two criteria are chosen to address the ambivalence between separation and exchange, as the main characteristic of a frontier.

We define the *occupancy* as the ratio between the number of cells forming the frontier and the total number of cells in the world:  $occupancy(F) = \frac{\#(A_F \cup V_F)}{\#A + \#V}$

For instance, if each agent is placed on a checkerboard according to its type<sup>1</sup>, all the agents are on the frontier and so the occupancy is equal to 1.

In Material Physics, *porosity* is a measure of how much of a rock is open space in between spores or within cavities of

<sup>1</sup>On a checker-board, a yellow agent is on a black square while a blue agent is on a white square, so there are no vacant places.



the rock  $\tau$ : it is defined as the ratio of the occupancy of voids in a material to the occupancy of the whole. By analogy, we consider the elements of  $D$  as representing the voids in a material (a lack of communication impediments). We define then the *porosity* as the proportion of direct-contacts: 
$$porosity(F) = \frac{\#D}{\#D + \#I}$$

### Simulation Framework

Experiments are performed *via* the *NetLogo* multi-agent programmable modelling environment  $\tau$ . The pseudocode for simulating AR models is defined in algorithm  $\tau\tau$ ; by instantiating the utility function *via* the update satisfaction method we obtain the six simulators that we experiment with.

Simulations are performed on a  $L \times L$  lattice of locations:  $L$  is set to 50 and the density of agents  $d$  is 90% which are standard values to simulate the Schelling model  $\tau$ . The agent set is positioned in a random initial configuration, such that the vacant locations and the two types of agents are well mixed and the *segMix* index is close to 0.

---

#### Algorithm 1 Simulation of the AR models

---

1.  $t \leftarrow 0$ ,  $density \leftarrow d$ ,  $threshold \leftarrow \tau$
  2. create a grid network and position at random the agents on it
  3. update the satisfaction of all the agents at time 0
  4. **while** not (all the agents are satisfied) **do**
  5.   **for** each agent  $a_i$  **do**
  6.     **if** not ( $a_i$  satisfied) **then**
  7.       choose a node-location at random on the grid
  8.       **if** the location is vacant **then**
  9.          $a_i$  moves to this location
  10.      **end if**
  11.    **end if**
  12.   **end for**
  13.    $t \leftarrow t + 1$
  14.   update satisfaction of all the agents at time  $t$
  15. **end while**
- 

### Models of segregation

In this section we consider among the six AR models the ones which lead to segregation. In order to compare the resulting shapes we will use (i) the *segMix* index, (ii) the  $mM$  congruence index and (iii) the characteristics of the frontier which emerges between clusters of agents with a same type.

#### $M_0$ model

In the Schelling's segregation model, agents are satisfied if the proportion of dissimilar neighbours among the agent-neighbours is below a threshold and unsatisfied if the proportion of similar neighbours is above this threshold. The

Schelling's model  $\tau$  is a particular case among the AR models: it corresponds to  $M_0$ . To compute satisfaction, we consider the condition ( $\Delta \leq \tau$ ) where  $\tau$  can be interpreted as the *tolerance* of the agents against their dissimilar neighbours. If  $\tau \ll 0.5$ , agents are rather intolerant, else if  $\tau \gg 0.5$  agents are rather tolerant.

Because the smallest step in the fraction of satisfied neighbours is  $1/8$ , we consider the two examples with  $\tau = 3/8$  and  $\tau = 5/8$ : if  $\tau = 0.375$ , agents are rather intolerant: one agent  $a_i$  will be satisfied in eighteen cases. If moreover the agent has exactly eight nearby agents, it cannot suffer more than three dissimilar neighbours. At the opposite, for the value  $\tau = 0.625$  all the individuals are rather tolerant and, if moreover an agent has exactly eight nearby agents, it can suffer at most five dissimilar agents in its vicinity.

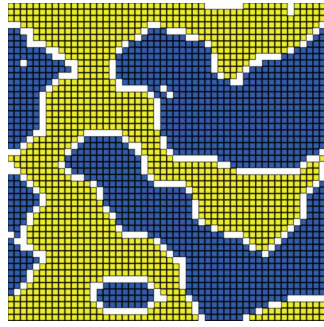
**segregation** Because  $M_0$  is self-dual, the dynamics can be interpreted either as a phenomenon of *repulsion* against dissimilar neighbours or *attraction* for similar neighbours. From the first point of view  $\tau$  is the *tolerance* against dissimilar while in the second case  $(1 - \tau)$  is the *appeal* for similar. Simulations show that both intolerant and tolerant agents lead the population to segregate with, of course, an *segMix* index higher for intolerant agents (0.952 vs. 0.519) (figure  $\tau\tau$ ). As we use the ER rule, this result confirms the unexpected behaviour provided by the Schelling's model where, in spite of their tolerance, to some extent, agents tend to group together by affinity.

**Micro-Macro congruence** The gap between the threshold of tolerance and the mean utility over the whole population is surprisingly high at the end of a run. If agents are intolerant ( $\tau = 0.375$ ) the congruence between micro and macro levels is close to 0 ( $mM = 0.07$ ); in this way, complex dynamics build much more liveable configurations than necessary. The fact that tolerant agents ( $\tau = 0.625$ ) tend nevertheless to group together by affinity can be explained by a low micro-macro congruence ( $mM = 0.37$ ).

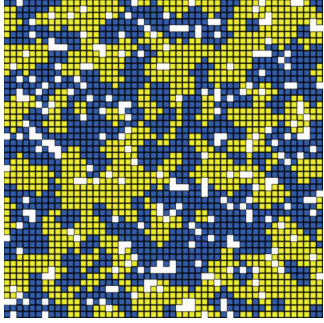
**Frontier** If agents are intolerant, the emerging frontier is essentially built with vacant cells (figure  $\tau\tau$ : white squares). There are many indirect-contacts and homogeneous patterns are isolated by a *no-man's-land* of vacant-cells. As tolerance increases the *no-man's-land* shape becomes more and more complex: as in a real landscape when roughness dictates many meanders to the edge of a lake, the complexity of contours increases (figure  $\tau\tau$ ). We observe that both the *occupancy* and the *porosity* increase with tolerance. So, both the shape and the composition of the frontier change as agents become more and more tolerant.

#### $M_1$ model: repulsion for the unlike

The  $M_1$  model is a variant of the  $M_0$  model where agents are satisfied if the normalized number of dissimilar agent-



(a)  $\tau=0.375$   
 $\text{segMix}=0.952$   $mM=0.07$   
 $\text{occupancy}=0.22$   $\text{porosity}=0.55$



(b)  $\tau=0.625$   
 $\text{segMix}=0.519$   $mM=0.37$   
 $\text{occupancy}=0.69$   $\text{porosity}=0.91$

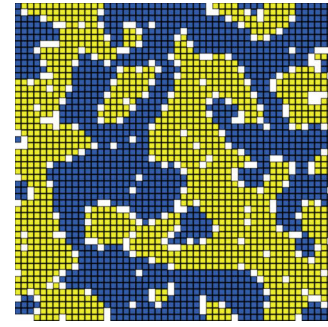
Figure 1:  **$M_0$  model:** repulsion for the unlike or attraction for the like.

neighbours is below a threshold. So, the threshold  $\tau$  represents the *tolerance* of the agents.

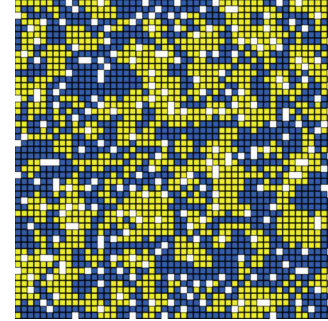
**segregation** The dynamics can be interpreted as a phenomenon of *repulsion* against dissimilar neighbours. Once again, this should lead the system to converge toward a global configuration with segregation. Indeed, we can observe that both intolerance (figure ??) and tolerance (figure ??) lead to segregation. Nevertheless, compared to the  $M_0$  model, the phenomenon of segregation is less marked (table ??); more, if agents are rather tolerant ( $\tau = 0.625$ ), this tendency to group together by affinity is much lower than in the  $M_0$  model ( $\text{segMix} = 0.3$  vs.  $\text{segMix} = 0.519$ ).

**Micro-Macro congruence** For intolerant, as well tolerant, agents the gap between the threshold and the mean utility over the whole population is much lower than in the  $M_0$  model (e.g.  $mM = 0.23$  vs.  $mM = 0.07$ ).

**Frontier** As tolerance increases, there is always a tendency to transform *no-man-land* frontier to *space-fill* curve; but here the frontier occupies more places in the world and its porosity is higher (figure ??).



(a)  $\tau=0.375$   
 $a=0.814$   $m=0.23$   $o=0.40$   $p=0.78$



(b)  $\tau=0.625$   
 $a=0.3$   $m=0.51$   $o=0.88$   $p=0.9$

Figure 2:  **$M_1$  model:** repulsion for the unlike.

### $M_1^*$ model: attraction for the like

As the  $M_1^*$  model is the dual of  $M_1$ , micro-attraction against similar agents-neighbours induces macro-segregation. Indeed, agents are satisfied if the normalized number of similar agents-neighbours is above the threshold: so,  $\tau$  represents the *appeal* of one agent for its similar neighbours. Let's note that the more the *appeal* the more the incentive to move is. As, at the local level, there is *attraction* for the cells which share the same type, once again, this should lead the system to converge toward a global configuration with high segregation. Let's remark that in order to compare the macro behaviour between models  $M_1$  (or  $M_0$ ) and  $M_1^*$ , we have to consider *tolerance*  $\tau$  with *appeal*  $(1 - \tau)$  (e.g. 0.375 vs. 0.625).

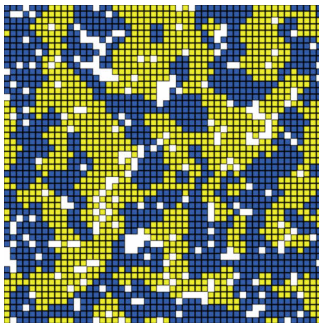
**segregation** The emerging segregation can be interpreted as *homophily*, that is the tendency of agents to segregate in spatial groups with similar others. First of all, let's note that the system *converges* towards global satisfaction if  $\tau = 0.375$ ; but, with  $\tau = 0.625$ , it remains forever about 10% of unsatisfied agents in the population; so there is *quasi-convergence* only. As expected, we can observe that high appeal (e.g.  $\tau = 0.625$ ) leads to a strong segregation (e.g.  $\text{segMix} = 0.91$ ) (figure ??); but, more surprisingly, low appeal (e.g.  $\tau = 0.375$ ) leads also to some extend to an

segregate population of agents ( $segMix = 0.567$ ) (figure ??) comparable to what we obtain for the  $M_0$  model with tolerant agents.

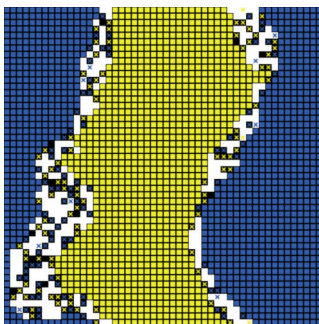
**Micro-Macro congruence** For appealing agents the gap between the threshold and the mean utility over the whole population is much lower than in the  $M_0$  model ( $mM = 0.22$  vs.  $mM = 0.07$ ); results are rather comparable to what we obtain for the  $M_1$  dual model ( $mM = 0.22$  vs.  $mM = 0.23$ ).

The fact that agents weakly *attracted* by similar neighbours segregate nevertheless in spatial group with similar others can be explained by a low micro-macro congruence ( $mM = 0.37$ ) comparable to what we obtain for the  $M_0$  model with tolerant agents.

**Frontier** When the strength of attraction towards similar neighbours is high, the frontier have stable properties (i.e. occupancy and porosity are quasi-invariant). Moreover, these properties are comparable to what we obtained with the  $M_0$  model (Table ??). Nevertheless, due to its permanent dynamics, the frontier have now a radically different shape: there is a certain thickness inside which the cells move infinitely (figure ??).



(a)  $\tau=0.375$   
 $a=0.567$   $m=0.47$   $o=0.66$   $p=0.92$



(b)  $\tau=0.625$   
 $a=0.91$   $m=0.22$   $o=0.20$   $p=0.61$   
 $s=90\%$

Figure 3:  $M_1^*$  model: attraction for the like.

## Models of mixing

In this section we consider the models which lead to mixing among the AR models. In these cases it is useless to look at the frontier because, at convergence, this one occupies the world in totality (i.e.  $occupancy \approx 1$ ) and porosity is very high (i.e.  $porosity \approx 0.9$ )

### $\overline{M}_0$ model

In this model agents are satisfied if the proportion of dissimilar neighbours is above  $\tau$  and unsatisfied if the proportion of similar neighbours is below  $\tau$ . Let's remember that this model is self-dual. Because it is the complement of  $M_0$ , macro-mixing is induced either by micro-repulsion against similar neighbours or micro-attraction by the agent-neighbours which share the same type.

**Mixing** We can observe indeed that high value of  $\tau$  leads to a low value of  $-0.53$  for the  $segMix$  index which reveals a strong mixing: the population is mainly an alternation of homogeneous lines or columns constituted by agents with the same type (figure ??). As  $\tau$  decreases, the  $segMix$  index increases too and so mixing decreases (figure ??).

**Micro-Macro congruence** Whatever the threshold is, the gap between the threshold and the mean utility over the whole population is relatively low at the end of a run ( $mM \approx 0.64$ ).

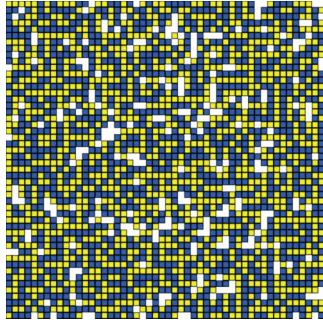
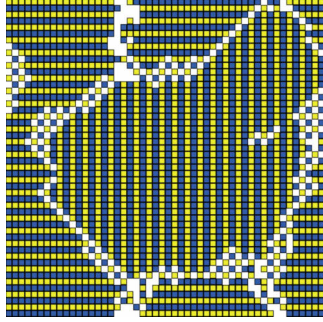
### $\overline{M}_1$ model: attraction for the unlike

In the  $\overline{M}_1$  model agents are satisfied if the normalized number of dissimilar agent-neighbours is above the threshold. Because this model is the complement of  $M_1$ , micro-attraction for dissimilar neighbours will induces macro-mixing.

**Mixing** First of all, let's note that the system *converges* towards global satisfaction if  $\tau = 0.375$ ; but, with  $\tau = 0.625$ , it always remains about 40% of unsatisfied agents in the population; so there is *quasi-convergence* only. In spite of the quasi-convergence phenomena, we can observe that a high value of  $\tau$  leads in the long term to a low value around  $-0.306$  for the  $segMix$  index which reveals mixing: the satisfied agents are then organized as an alternation of homogeneous lines or columns constituted by agents with the same type (figure ??). As  $\tau$  decreases, the  $segMix$  index increases too and so mixing decreases (figure ??).

**Micro-Macro congruence** The gap between the threshold of tolerance and the mean utility over the whole population is surprisingly low at the end of a run (if  $\tau = 0.375$ ,  $mM = 0.73$  and if  $\tau = 0.625$ ,  $mM = 0.96$ ); in this way, complex dynamics build needed liveable configurations only.

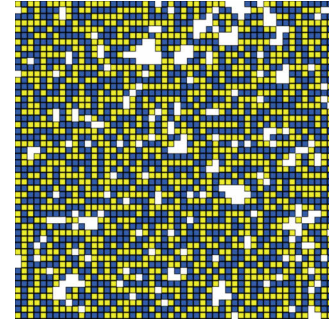
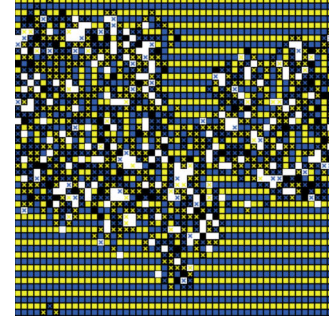
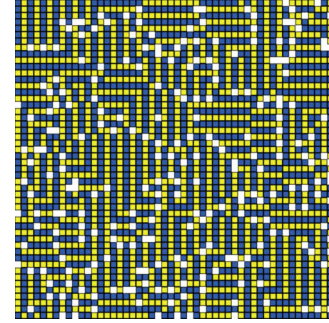
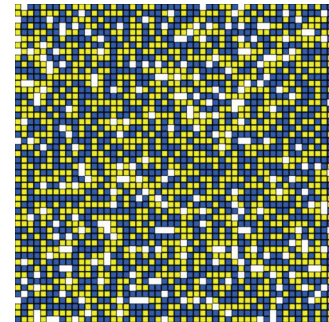


(a)  $\tau=0.375$   $a=-0.183$ ,  $m=0.65$ (b)  $\tau=0.625$   $a=-0.53$ ,  $m=0.63$ Figure 4:  $\overline{M}_0$  **model**: attraction for the unlike or repulsion for the like. **$\overline{M}_1^*$  model: repulsion for the like**

In the  $\overline{M}_1^*$  model agents are satisfied if the normalized number of dissimilar neighbours is below the threshold. Because this model is the complement of  $M_1^*$ , micro-repulsion for similar neighbours will induces macro-mixing for low value of  $\tau$ .

**Mixing** Simulations show that the system converges toward a population of satisfied agents where a low value around  $-0.421$  for the *segMix* index reveals a strong mixing: the satisfied agents are mainly organized as an alternation of homogeneous lines or columns constituted by agents with the same type (figure ??). Let's note that the vacant cells are placed at the crossroad between line and column and so allow right or left turns in the structure. As  $\tau$  increases, the *segMix* index increases to a value close to zero and so mixing disappears (figure ??).

**Micro-Macro congruence** There is high congruence between micro-motive and macro behaviour at the end of a run (if  $\tau = 0.375$ ,  $mM = 0.69$  and if  $\tau = 0.625$ ,  $mM = 0.66$ ); in this way, complex dynamics build needed liveable configurations almost only.

(a)  $\tau=0.375$   $a=-0.189$ ,  $mM=0.73$ (b)  $\tau=0.625$   $a=-0.306$ ,  $m=0.96$   $s=60\%$ Figure 5:  $\overline{M}_1$  **model**: attraction for the unlike(a)  $\tau=0.375$   $a=-0.421$ ,  $m=0.69$ (b)  $\tau=0.625$   $a=-0.081$ ,  $m=0.66$ Figure 6:  $\overline{M}_1^*$  **model**: repulsion for the like.



## Discussion and conclusion

Taking inspiration from the Schelling's segregation model, we have proposed a family of models to study the effects of various micro-motives based on attraction or repulsion on the emergence of macro segregation or mixing. Table ?? summarizes results obtained from simulations; all the values are averaged over 100 independent runs.

The *segregate-mixing* index reveals: (i) three expected cases with strong segregation (bold values), (ii) three expected cases with strong mixing (bold values), (iii) two surprising cases with segregation: tolerant agents in the Schelling- $M_0$  model and unappealing agents in the  $M_1^*$  model lead to macro-segregation (underline values), (iv) no unexpected case with mixing.

The *micro-Macro* index reveals: (i) for all the mixing models the micro-macro congruence is high (above 0.6) (ii) for all the segregation models the micro-macro congruence is low (below 0.5) (iii) two extreme cases (bold values): intolerant agents in the  $M_0$  model lead to build much more liveable local configurations than necessary and appealing agents in the  $\overline{M}_1$  model lead to build needed liveable configurations only.

Measuring *occupancy* of frontier reveals that strong segregation comes with low occupancy (bold values).

Measuring *porosity* of frontier reveals: (i) high porosity emerges from tolerant or unappealing agents (bold values), (ii) the lower porosity comes with strong segregation.

We now attempt to provide elements to respond to the issues formulated in introduction.

$Q_1$  Comparison between  $M_0$  and  $M_1$  (or  $\overline{M}_0$  and  $\overline{M}_1$ ) shows that it is not equivalent to flee regarding the *proportion* of dissimilar agents among the neighbourhood agents rather the *number* of dissimilar agents in the full set of neighbours.

$Q_2(a)$  Because  $M_0$  (resp.  $\overline{M}_0$ ) is self-dual, it is equivalent to be attracted by similar (resp. dissimilar) or repulsed by dissimilar (resp. similar) if each one is influenced by the *proportion* of similar among its neighbours.

(b) Because  $M_1$  and  $M_1^*$  (or  $\overline{M}_1$  and  $\overline{M}_1^*$ ) are dual and the respective macro behaviours are different, it is not equivalent to be attracted by similar (resp. dissimilar) rather to be repulsed by dissimilar (resp. similar) neighbours if each one is influenced by the *number* of similar among its neighbours.

Table 4: Models of segregation / mixing

	$\tau$	segMix	mM	occup.	poro.
$M_0$	0.375	<b>0.951</b>	<b>0.065</b>	<b>0.206</b>	0.524
	0.625	<u>0.527</u>	0.377	0.704	<b>0.915</b>
$M_1$	0.375	<b>0.801</b>	0.219	<b>0.425</b>	0.795
	0.625	0.298	0.506	0.895	<b>0.903</b>
$M_1^*$	0.375	<u>0.540</u>	0.488	0.689	<b>0.918</b>
	0.625	<b>0.902</b>	0.240	<b>0.194</b>	0.609
$\overline{M}_0$	0.375	-0.179	0.656	$\approx 1$	$\approx 0.9$
	0.625	<b>-0.528</b>	0.628	$\approx 1$	$\approx 0.9$
$\overline{M}_1$	0.375	-0.189	0.727	$\approx 1$	$\approx 0.9$
	0.625	<b>-0.296</b>	<b>0.951</b>	$\approx 1$	$\approx 0.9$
$\overline{M}_1^*$	0.375	<b>-0.425</b>	0.683	$\approx 1$	$\approx 0.9$
	0.625	-0.084	0.652	$\approx 1$	$\approx 0.9$

# Molecular robotics approach for constructing an artificial cell model

Shin-ichiro M. NOMURA<sup>1</sup>, Yusuke SATO<sup>1</sup> and Kei FUJIWARA<sup>1,2</sup>

<sup>1</sup> Tohoku University, 6-6-01, Aramaki Aza-aoba, Aoba-ku, Sendai, Japan

<sup>2</sup> JSPS Research Fellowship for Young Scientists  
nomura@molbot.mech.tohoku.ac.jp

## Abstract

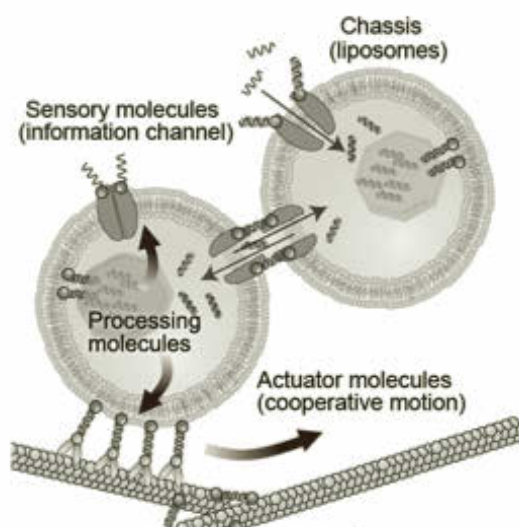
Prototype artificial cell models with designed functional molecules are presented here. Artificial molecular devices based on a giant liposome were prepared to obtain specific properties that cannot be obtained from natural cells. In this context, artificial cell research is seen an extension of “molecular robotics” research. Cooperative and integrated chemical systems will be constructed from the molecular devices. Here, we present the 3 aspects of the study model: (1) gene-expressing cell model encapsulated in the liposome to simulate membrane protein synthesis, (2) multirole molecular device with a designed DNA nanostructure on the cellular membrane, and (3) designed membrane peptide device for surface recognition. Although these devices are inspired by living cell functions, such goal-oriented systems are free from the constraints of natural history and evolution. These artificial devices may be integrated to develop novel tools for living systems.

## Introduction

All living organisms are composed of cells, and cells are constructed from various molecules. Since the end of the last century, rapid progress in molecular science and bioengineering has enabled the analysis of complex living phenomena at a molecular level. Such a top-down approach has provided essential pictures of molecular systems, such as the entire human genome, proteome, or metabolome. However, constructive research has also been essential and is already used for evaluating biochemical reaction systems. Such a bottom-up approach also aims to build basic molecular systems from individual molecules. The goal of artificial cell research to create a cell-like structure in a spatiotemporal manner by using a designed molecular system[1]. Several research groups have reported to construct artificial cell models by using liposomes, that encapsulate biochemical reaction networks such as gene expression from a template DNA [2]. In the last decade, synthetic biology has been the main constructive approach, and living cell functions have been modified using standardized genetic components. The establishment of induced pluripotent cells [3] and total synthesis of the bacterial genome [4] are great milestones of this research area.

The goal of the bottom-up and top-down approaches is to construct the entire cell at the molecular level. Such realistic artificial cell studies seek to reproduce the history of the cell at the molecular level and may thus

address the origins of life itself. However, we have also noted the “artificial” aspect of such cell models. It would be possible for the artificial structure in such models to perform a difficult task that is impossible for living cells. Such a capability might contribute to research in a different way from genetically modified cells.

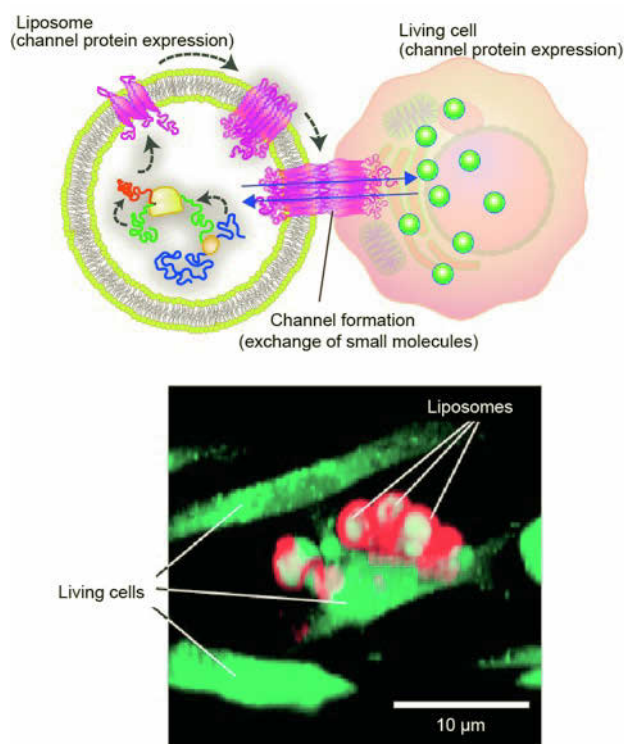


**Fig. 1.** Schematic illustration of a possible artificial cell model consisting of a designed molecular system. We call this system “molecular robotics.”

We are also creating models with designed functional molecules. Such non-natural molecules confer specific properties (e.g., sensing, actuation, and computing) to the artificial cell compartment, which is a liposome. Such artificial molecular devices can be integrated into a complex molecular system to provide a “molecular robot” [5]. In such a context, artificial cell research is included as a subset of molecular robotics (Fig. 1.)

The study model has 3 aspects: (1) a gene-expressing cell model encapsulated in the liposome to enable membrane protein synthesis, (2) a multirole molecular device with a designed DNA nanostructure on the cellular membrane. (3) a designed membrane peptide device for surface recognition. Although these devices are inspired by living cell functions, such goal-oriented systems could become free from the constraints of natural history.

## Gene expression in an artificial cell



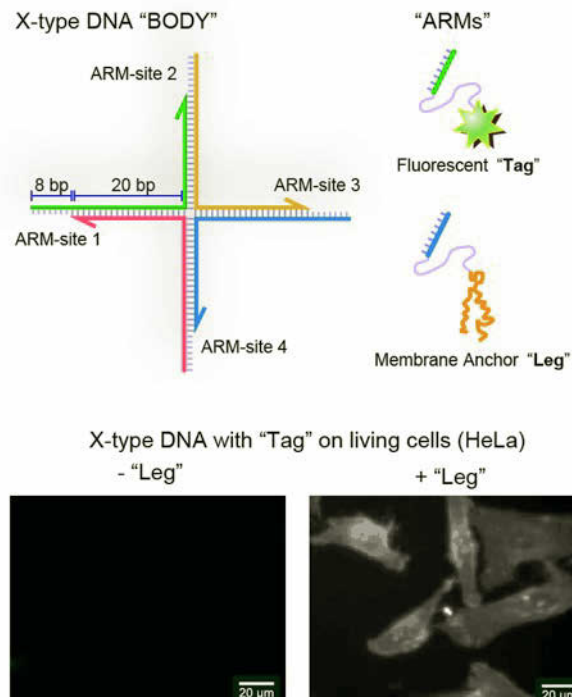
**Fig. 2.** Artificial cell model liposomes containing a gene expression system. Upper: Schematic illustration of the system. Lower: The membrane protein connexin was expressed and functioned on the liposome membrane. Liposomes were observed inside cultured cells.

We have constructed a series of artificial cell models based on giant liposomes. A giant liposome is a spherical structure that consists of a closed lipid bilayer membrane, with a diameter greater than several micrometers. The giant liposome membrane is known as the simplest model of the living cell membrane [6]. Several protein synthesis reactions with coupled transcription and translation have been reported by introducing various kinds of functional molecules into liposomes [2, 7, 8]. Expression of functional membrane proteins in the liposome has also been reported in successful [2f, 8, 9]. Connexin-containing liposomes were prepared by using a cell-free transcription/translation system with a plasmid encoding connexin in the presence of liposomes. The nascently expressed membrane protein, connexin, was directly constituted to the liposome membrane on performing in vitro transcription and translation, thus generating pure membrane protein-containing liposomes. The hydrophilic dye calcein or peptides were efficiently transferred from connexin-expressing liposomes to cultured cells (Fig. 2).

One of our future goals in the expansion of this approach is total reconstitution of a living cell. Using extracted cell components from cultured cells, we are trying to completely reconstruct cellular components under conditions approximating those of living cells [10]. We adopt elemental molecular complexes without further processing if they are functional. These approaches may be termed as middle-out approaches. Such studies should indicate how functional components can be managed to obtain a complex life-like system.

## Multirole molecular device based on designed DNA

Recently, DNA has been used as a programmable building material through self-assembly in DNA nanotechnology. Several methods have been proposed for the construction of nanostructures from DNA molecules such as DNA tiles [11], DNA origami [12], and DNA bricks [13]. We can design static nanostructures by using computer-assisted design software (e.g., caDNAno (<http://cadnano.org>)). We designed an artificial molecule that can be used to attach exchangeable molecular devices to the cellular membrane. The X-type body consists of 4 individual single-stranded DNA (ssDNA) molecules with sticky ends, called “ARM sites.” Molecular device attachments are designed to be complementary with ARM-site DNA sequences. The unit called the “ARM” can be attached to the corresponding ARM site of the DNA sticky end.



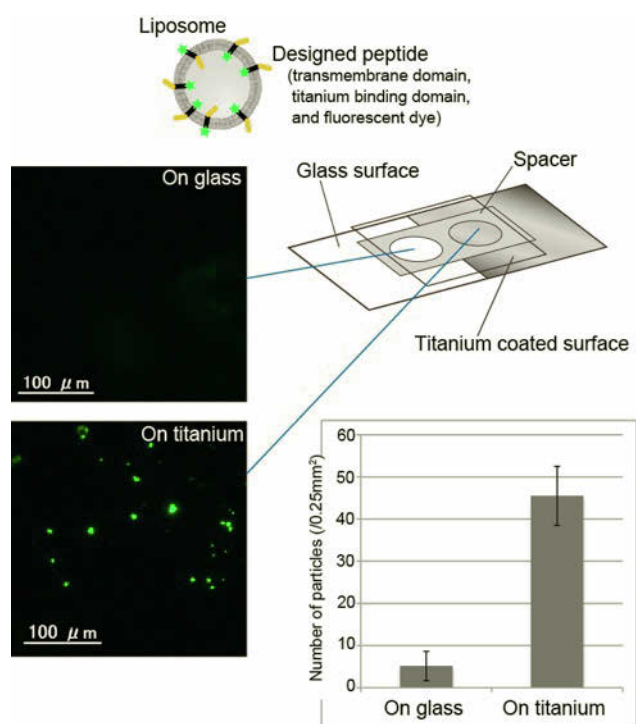
**Fig. 3.** X-type designed DNA molecules. Upper panel: schematic illustration of the molecular system. Lower panel: X-type DNA equipped with a “tag” was added to living human cultured cells.

The ssDNA sequence was designed for attachment while mixing at room temperature. “Legs” formed from hydrophobic molecules can also be used to attach the DNA body to the cellular membrane (Fig. 3, lower). The DNA body was stained using a “tag” ARM and was thus found to move on the cell surface by two-dimensional diffusion. The diffusion constant was investigated by single-molecular tracking on the cell surface and found to be approximately  $0.3 \mu\text{m}^2/\text{s}$ . The lipid bilayer is a universal molecular structure of the cell, and every receptor is located on a membrane. Similar to membrane proteins, membrane-localized molecular robots (molbots) may be used to control molecular information and compartmentalized conditions inside the cell. Simple molecular robots made of nucleic acids (DNA or RNA) such as those in the present study may be expressed in living cells by genetic engineering. When the molbot was appropriately designed with regard to  $T_m$  (melting temperature of double-stranded DNA or RNA), *in situ* production and function in the desired cell were achieved.

### Designed membrane peptide device for surface recognition

As described in the previous section, nucleic acids such as DNA or RNA are attractive molecules for prototyping of molecular devices. If other types of biomolecules can also be designed easily, they will be useful for the construction of artificial functional cell models. Proteins are the main functional component of organisms and occupy over 16 wt% of the total mass of the human cell.

Compared to DNA nanostructure design, protein design is difficult because protein function depends not only on the linear sequence but also on folding states and post-translational modifications. However, small units of protein, i.e., peptides, can be easily designed and are easy to obtain as commercial molecules. Water-soluble peptides are commonly used as drugs or in the cosmetic field. Here, we aimed to design an artificial sensory molecule for attachment to the liposome. The transmembrane  $\alpha$ -helix domain was designed based on a previous report [14]. A functional metal (Ti)-specific binding domain was prepared using a procedure reported in a previous study [15]. The designed amino acid sequence of the peptide can recognize the specific electrostatic potential of a metal surface and then bind to the surface. The designed amphiphilic peptide was also attached to a fluorescent molecule and mixed with a lipid solution (1 mM DOPC:DMPC:Chol = 6:1:2 with 50 nM peptide) to form a modified liposome with a diameter of 200 nm. The sample solution was placed onto glass with or without a titanium coating. Fluorescence microscopic observation clearly showed that the designed peptide embedded in the liposome membrane could attach to the titanium surface (Fig. 4). Functional designed peptides should also be synthesizable by gene expression in the giant liposome. Construction of a trigger system for expression control (e.g., riboswitches) is awaited.



**Fig. 4.** Surface attachment of artificial liposomes equipped with the designed membrane peptides.

### Conclusion

In this report, we have described our approach for constructing an artificial cell model, that is, encapsulation of biochemical reactants and artificially designed DNA and peptides. However, only combining the functional molecules can never give rise to functional structures; development of the molecular-processing system is a crucial step. If the model is compartmentalized, control of molecular input/output through the membrane is essential. To obtain reliable systems, the transduction mechanism needs to have both noise reduction and signal amplification. Implementation of multiple inputs and multiple outputs coupled with an internal chemical reaction network must also be considered. Given these critical issues, a self-reproducing system is a distant goal. Concerning about an artificial "cell", molecular robotics approach should also support an effort for a cell total reconstitution from natural materials [4, 10, 16]. Currently, undergraduate students have designed bacterial genetic circuits ([http://igem.org/Main\\_Page](http://igem.org/Main_Page)) and DNA nanostructures (<http://biomod.net>). Thus, the current progress in this field indicates that it should be possible to obtain new artificial cell models in the near future.

### Acknowledgements

We would like to thank Prof. S. Murata and S. Hamada for their insightful comments. This work was supported by



JSPS KAKENHI (grant numbers 24104004, 22220001, and 23.3718).

## References

- [1] Szostak J.W., Bartel D.P. and Luisi P.L.(2001), Synthesizing life, *Nature* **409**:387-390.
- [2] a) Yu W. *et al.*(2001), Synthesis of functional protein in liposome. *J. Biosci. Bioeng.*, **92**, 590-593; b) Kuruma Y., P. Stano P., Ueda U. and Luisi P. L.(2009), A synthetic biology approach to the construction of membrane proteins in semi-synthetic minimal cells. *Biochim. Biophys. Acta Biomembr.*, **1788**, 567-574; c) Kita H. *et al.*(2008), Replication of genetic information with self-encoded replicase in liposomes. *ChemBioChem*, **9**, 2403-2410; d) Sunami T., Hosoda K., Suzuki H., Matsuura T., Yomo T.(2010), Cellular compartment model for exploring the effect of the lipidic membrane on the kinetics of encapsulated biochemical reactions. *Langmuir*, **26**, 8544-8551; e) Pereira de Souza T., Stano P. and Luisi P.L.(2009), The minimal size of liposome-based model cells brings about a remarkably enhanced entrapment and protein synthesis. *ChemBioChem*, **10**, 1056-1063; f) V. Noireaux, A. Libchaber (2004), A vesicle bioreactor as a step toward an artificial cell assembly. *Proc. Natl. Acad. Sci. USA*, **101**, 17669-17674; g) Saito H. *et al.* (2009), Time-resolved tracking of a minimum gene expression system reconstituted in giant liposomes. *ChemBioChem*, **10**, 1640-1643; h) Nourian, Z., Roelofsen, W. and Danelon, C. (2012) Triggered Gene Expression in Fed-Vesicle Microreactors with a Multifunctional Membrane. *Angew. Chem. Int. Ed.* **51**, 3114–3118.
- [3] Takahashi K. and Yamanaka S.(2006), Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors, *Cell*, **126**:663-676.
- [4] Gibson, D. *et al.* (2010), Creation of a bacterial cell controlled by a chemically synthesized genome. *Science* **329**:52-56.
- [5] Murata, S., Konagaya, A., Kobayashi, S. and Saito, H. (2013), Molecular Robotics: A New Paradigm for Artifacts. *New Generation Computing*, **31**:27-45.
- [6] Walde, P., Cosentino, K., Engel, H. and Stano, P. Giant Vesicles: Preparations and Applications. *ChemBiochem* 848–865 (2010).
- [7] Nomura S.-i.M. *et al.* (2003), *ChemBioChem*, **4**:1172-1175.
- [8] Nomura S.-i.M. *et al.* (2008), *J. Biotechnology*, **133**:190-195.
- [9] Kaneda, M. *et al.* (2009), *Biomaterials*, **30**: 3971-3977.
- [10] Fujiwara, K. and Nomura, S.-I. M. (2013), Condensation of an additive-free cell extract to mimic the conditions of live cells. *PLoS ONE* **8**: e54155.
- [11] Winfree, E., Liu, F., Wenzler, L., Seeman, N. (1998), Design and self-assembly of two-dimensional DNA crystals, *Nature*, **394**:1539–544.
- [12] Rothmund, P. (2006), Folding DNA to Create Nanoscale Shapes and Patterns, *Nature*, **440**:297–302.
- [13] Yonggang K., Luvena L.O., Shih W.M., Peng Y. (2013), Three-Dimensional Structures Self-Assembled from DNA Bricks, *Science* **338**: 1177-1183.
- [14] Yano Y. *et al.* (2002). Topological stability and self-association of a completely hydrophobic model transmembrane helix in lipid bilayers. *Biochemistry*, **41**:3073–3080. [15] Sano K., Shiba K. A.(2003) hexapeptide motif that electrostatically binds to the surface of titanium. *J. Am. Chem. Soc.*, **125**: 14234-14235. [16] Fujiwara, K., Katayama, T. &

Nomura, S.-i. M.(2013) Cooperative working of bacterial chromosome replication proteins generated by a reconstituted protein expression system. *Nucleic Acids Res.* (2013). doi: 10.1093/nar/gkt489

# Lévy-like Distribution Shown by Intermittent Search Model with Misunderstanding Switch Pattern

Hisashi Murakami<sup>1</sup> and Yukio-Pegio Gunji<sup>1</sup>

<sup>1</sup>Kobe University  
hssh415@gmail.com

## Abstract

In an intermittent random search, in which slow motion to detect the target is discretely separated from the motion to migrate to another feeder, the high efficiency of the Lévy strategy is generally found, meaning that the time interval of phase switching is chosen from the Lévy distribution. Though the Lévy strategy is consistent with the searching behavior of real animals, some researchers claim that the Lévy-like distributions exhibited by animals are not necessarily produced by a Lévy process. Here, we propose an intermittent two-phase search model that does not include a Lévy process. In this model, the agent is basically a correlated random walker (CRW), but it memorizes its trajectory and counts the number of crossovers in a trajectory. If the number exceeds a threshold, the agent resets the memory of trajectories and makes ballistic movement in the direction uncorrelated to the past. We also show that this model can optimize the trade-off between macro search (exploration) and micro search (exploitation), which is shown by the CRW. Finally, we demonstrate that another intermittent search model that uses an ambiguous rule to switch the two phases can show a Lévy-like distribution of time intervals.

## Introduction

It is interesting to try to understand how living organisms navigate to targets in a natural environment, where resources are usually unpredictably distributed such that there is limited information about their locations (Viswanathan et al., 2011). The Lévy walk (LW) is considered the most important model of this type of random search, which is a special random walk in which each step length is chosen from a power-law distribution with a heavy tail (a so-called Lévy distribution) (Viswanathan et al., 2008, Reynolds and Rhodes, 2009). The LW shows a scaled step length  $l$  such that  $P(l) \sim l^{-\mu}$  with  $1 < \mu < 3$ , where  $\mu$  represents the power-law exponent. In the foraging simulations, if prey is abundant and, thus, predictable, it is known that classical random walks such as Brownian motion can yield higher encounter rates than LW. In contrast, when preys are sparsely and unpredictably located, LW is more efficient than classical random walks.

In this sense, LW has a reliable theoretical advantage, but is it consistent with empirical data? Indeed, it has been reported that, among diverse organisms, experimental evidence of LW can be found (Humphries et al., 2010, Cole, 1995, Viswanathan et al., 2008). It is most evident in the wandering

albatross. In the first of a series of pioneering works by Viswanathan et al., albatross behavior was tracked by using a humidity sensor attached to one leg of each bird (Viswanathan et al., 1996). Flight-time intervals were measured by wet periods, and dry points were considered to be landings on the water to catch fish. A reinvestigation using GPS, however, showed that most long flights in fact consisted of rest time, when the bird was in its nest, and concluded that there is no power-law distributed step length in wandering albatross (Edwards et al., 2007). Nevertheless, the latest study, which used the same method as above but examined the birds one by one, found that individuals foraging for sparse food exhibited certain Lévy movement patterns eventually (Humphries et al., 2012).

A trajectory of LW describes a search pattern composed of many small step clusters, interspersed by longer relocations known as “saltations” (O’Brien et al., 1990). This pattern can be intuitively described as an intermittent random search strategy in which slow motion is used to detect the target and a discretely separated motion is used to migrate to another feeding location (Bénichou et al., 2011). For example, if we lose a tiny object (e.g., a key) in a huge field, we can consider two simple ways to detect the key: a slow, careful search and a rough, fast one. In the former case, we can search accurately, but we would spend a very long time in the field. In the latter case, we may detect the key quickly, but in many cases, the lack of accuracy would result in just as long of a search time as the slow search. This illustrates a trade-off between the exploitation of old certainties and the exploration of new possibilities, which is frequently found in biology (March, 1991). To balance this dilemma, we would eventually choose a combined strategy, i.e., an intermittent strategy.

In studies of the intermittent strategy, it is difficult to determine the optimal way to switch between the different motions. The Lévy strategy also plays an important role. Bartumeus and his colleague compared a correlated random walk (CRW) (Kareiva and Shigesada, 1983), which is known as a natural way to model the emergence of angular correlations in animal trajectories coming from local scanning, with an intermittent model based on a CRW but incorporating uncorrelated reorientations with a time interval whose length is chosen from the Lévy distribution (Bartumeus and Levin, 2008). Then, they showed that this Lévy intermittent model is more efficient than the non-intermittent version.

Through these theoretical and empirical studies, the Lévy strategy has been established as a key to understanding animal search behavior. Consequently, it has been proposed that the Lévy strategy must be a strong target for natural selection. This is the so-called Lévy foraging hypothesis (Viswanathan et al., 2011). Still, some researchers claim that the Lévy-like distribution shown by animals is not necessarily produced by a Lévy process. Indeed, few models in which an agent walks deterministically and interacts with complex distributed targets can show an LW pattern (Santos et al., 2007). Moreover, Benhamou used combined exponential distributions to suggest that there is no guarantee that a Lévy-like distribution is based on a Lévy process (Benhamou, 2007).

Here, we will show a simple intermittent model that is not based on a Lévy distribution but does possess the principal features of an intermittent strategy, i.e., it shows two different phases. In this model, the number of crossovers in a trajectory is regarded as the extent to which the agent implements local search, and it also represents the threshold used to switch between the two phases. We demonstrate how this model can strike a balance in a trade-off between macro search (exploration) and micro search (exploitation), and we compare the model with a CRW. Finally, we describe another intermittent search model that uses an ambiguous switching rule. The ambiguity results from a stochastically generated long trail, and it generates a search in which the agent wastes too much time. Moreover, we demonstrate that the model can show a Lévy-like distribution of the time intervals.

## Results

### Basic models

First, we present a simple model that includes the main factor of intermittent search, in which an agent iterates to form local scanning behavior (here, we call this the exploitation phase). The local scanning is interspersed by longer relocations or saltations (exploration phase) in continuous space and discrete time. We refer to the entire model as EERW for short.

In the exploitation phase, the agent basically moves as a CRW. Here, angular correlations are introduced on the basis of a circular Gaussian distribution ( $-1.0 \leq g \leq 1.0$ ) centered at the value  $g = 0$  (maximum probability), although other distributions (e.g., a wrapped Cauchy distribution) might be as good (Bartumeus and Levin, 2008). The turning angle is represented by  $\theta = g\pi$ . At each step, the angle of the agent is determined by combining the turning angle with the previous angle. The standard deviation ( $SD$ ) of the Gaussian distribution controls the directional persistence or correlation length of the random walk (Bartumeus et al., 2005, Viswanathan et al., 2005).

In intermittent models, the agents should migrate to another feeder if the local scanning is finished. Hence, in EERW, the number of crossovers in a trajectory represents the extent to which the agent has searched the surrounding area, although switching between phases has usually been implemented by means of a stochastic process such as the Lévy distribution in previous models (Bartumeus and Levin, 2008). In other

words, each agent memorizes its trajectory and counts the number of crossovers in the trajectory. The threshold number of crossovers needed to switch between the two phases is represented by  $NC$ . Note that  $NC$  can be reasonably estimated as the extent of the local search, as we discuss later. If the number of crossovers exceeds the  $NC$  threshold, the agent makes a ballistic movement in a direction uncorrelated to its past, i.e., the turning angle is chosen from a uniform distribution  $\theta \in [-\pi, \pi]$ . Ballistic movement is continued until the time steps are proportional to the steps of the exploitation phase. The longer the agent stays in the exploitation phase, the larger the area searched by agent is, so the agent should spend as much time on local search as it does on relocation. The proportionality constant is represented by  $P$ . Because of its finite memory, the agent resets its memory of the trajectories, returns to the exploitation phase and starts walking again as a CRW at the new location.

The parameters in our model are listed below:

$SD$ : standard deviation of the Gaussian distribution with respect to directional persistence

$NC$ : threshold number of crossovers in a trajectory

$P$ : proportionality constant with respect to distance in the exploration phase

$l$ : step length

In this paper, we fixed the step length at  $l=0.5$ .

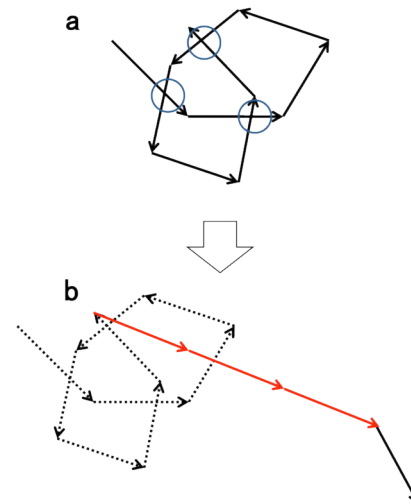


Figure 1 Schematic diagram of phase transition in the EERW model: (a) Steps of the Exploitation phase are represented by black arrows, and crossovers are surrounded by blue circles. (b) Steps of the Exploration phase are represented by red arrows, and steps of the previous and following Exploitation phases are represented by dashed and solid black arrows, respectively.

Fig. 1a shows the procedure from exploitation phase to exploration phase. In this case, the agent implements the exploitation phase as a CRW, and its trajectory has three crossovers, i.e.,  $NC = 3$ . When the agent switches to the exploration phase, it makes a ballistic movement comprising three steps, so  $P$  has a value of approximately 0.33 because there are nine steps spent in the exploitation phase.

Fig. 2 shows a series of snapshots of whole trajectories of the CRW and EERW models at  $T = 1000-100000$  with  $SD = 0.3$ ,  $l = 0.5$ ,  $NC = 10$  and  $P = 0.3$ . These simulations are implemented in continuous 2D space with no boundary conditions, but in Fig. 2, they are displayed as if they were in a space of  $200 \times 200$  with a wrapped boundary. It is easy to see that the search area of the EERW is broader than that of the CRW, even though they have the same search steps. In many cases, the agent with the search tasks has a perceptual range with a certain radius, in which the agent can detect a target (Viswanathan et al., 2011). Then, the search efficiency is estimated by the number of targets captured in this range. We may partially regard the search areas as representative of search efficiency. However, if the search area were the most important factor in a random search, it is strange that ballistic movement would be the most optimal strategy. In the next section, we show that EERW can balance a trade-off between macro search (exploration) and micro search (exploitation), which is shown by the CRW.

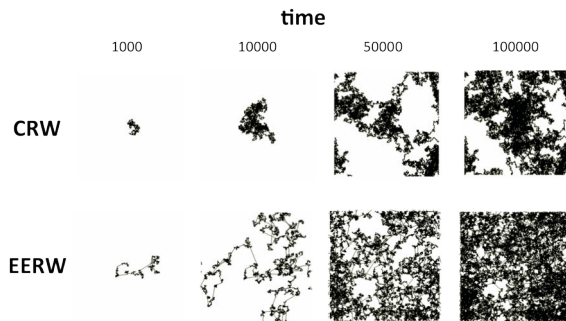


Figure 2 Snapshot of time development of the trajectories of the CRW (upper side) and EERW (lower side) in a continuous two-dimensional space of  $200 \times 200$  with wrapped boundary condition. Time proceeds from left to right. For both simulations,  $SD = 0.3$  and  $l = 0.5$ , and for EERW,  $NC = 10$  and  $P = 0.3$ . The trajectory of the CRW concentrates and overlaps at the center of the field, whereas that of the EERW is sparsely distributed but covers the entire field.

### A trade-off between macro search and micro search

Biological systems, from individual organisms to groups of animals, are subject to a trade-off between exploitation and exploration at various levels. Especially in open environments, the decision of whether to stay in a known environment or explore a new environment is a difficult one (March, 1991, Gunji et al., 2011, Bénichou et al., 2011). In a random (intermittent) search, the agent would move to another field if the local search were finished. Therefore, the exploitation-exploration dilemma would correspond to the relationship between micro and macro search. In fact, as mentioned above, if the agent moves by means of a strategy leaning toward micro search, it would spend a long time searching a huge field. Moreover, even if the agent detects some targets, those targets cannot be the most abundant resource. On the other hand, if the agent moves by means of a strategy leaning toward macro search, it also takes a long time

to detect targets because of search inaccuracy. Hence, there is a trade-off between micro search and macro search.

In this sense, the classical random walk is the strategy most biased against micro search, whereas ballistic movement is the most biased against macro search. Now, we show that CRW displays a micro-macro search trade-off, because as the parameter  $SD$  approaches zero, the behavior of CRW gets closer to ballistic movement. Conversely, as  $SD$  becomes larger, the behavior approaches a classical random walk. Here, we estimate the extent to which an agent implements micro search by means of the number of total crossovers in a trajectory of 5,000 simulation time steps, and we estimate the extent to which the agent implements macro search by means of closure areas. The closure areas are measured by total neighborhood areas with radii  $r = 1.0$ , where each arrival point of the agent is centered. Overall, the trade-off between micro search, represented by the total number of crossovers, and macro search, represented by the closure areas, is easy to see (Fig. 3). Patterns of CRW are generated with no boundary condition by varying the parameter  $SD$  from 0.01 to 1.0 in increments of 0.01. By comparing the patterns of CRW with those of EERW, we see that EERW can balance exploitation with exploration, and the patterns of EERW are generated under the same conditions as those of CRW, except that  $NC = 15$  and  $P = 0.3$ .

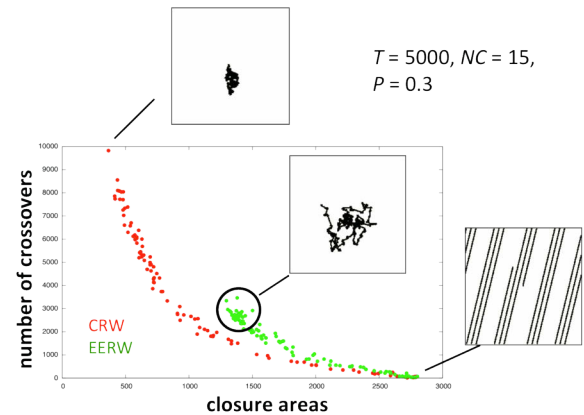


Figure 3 Performance of the CRW (red points) and EERW (green points) with respect to the closure areas and the number of crossovers. Snapshots of trajectories of the CRW with  $SD$  values of 0.0 and 1.0 are represented by top box and bottom box, respectively. Additionally, trajectories of the EERW with  $SD$  0.5 are represented by a box connected to a black circle, which indicates that the EERW can balance the trade-off.

Bartumeus and colleague showed that the Lévy intermittent model could be more efficient than the CRW (Bartumeus and Levin, 2008). In their study, the Lévy intermittent model was compared to the CRW model with two different values of a parameter that controls the directional persistence of the model. However, the CRW can display various behaviors, from persistent, ballistic movement to an uncorrelated classical random walk. We examined the CRW with most of the possible  $SD$  values, revealing that the EERW intermittent model without a Lévy process can balance the trade-off of the



CRW between exploitation and exploration, as calculated by two quantities: number of closure areas and number of crossovers.

### Another intermittent search model

In the study of random search, some models have assumed that agents have memory or learning skills (Ferreira et al., 2012, MacNamara and Houston, 1987). Their memories and learning abilities are also assumed to be finite. However, there has been no attempt to assume ambiguity of memory or misunderstanding of learning. In this section, we first introduce ambiguity and/or misunderstanding in the form of a rule to switch between search phases in the EERW. Second, we show that such a modified EERW (MEERW) results in a Lévy-like distribution of time intervals for phase switching. Finally, we discuss the difference between LW and MEERW.

In the EERW model, the number of crossovers in a trajectory represents the extent to which the agent implements local search. Hence, there was a threshold number of crossovers,  $NS$ , at which the agent switched phases. In contrast, in the MEERW model, which in most ways is the same as the EERW model (Fig. 4a), the value of the threshold  $NS$  is dynamically varied by two types of misunderstanding of the rule.

One such misunderstanding occurs when a long trail is stochastically generated without enough crossovers. We regard it as a ballistic movement (Fig. 4b). If such the long trail is generated, it is assumed that the exploitation phase was already implemented, even though the number of crossovers did not exceed the threshold  $NS$ , and exploration phase is regarded as a trail entailing the reset of the memory of the trajectory. Then,  $NS$  is decremented by one because the agent misunderstands a shorter local search (and the generated trail) as a rule. We here define such a long trail as a series of tracks comprising  $N_t$  tracks, in which the inner product of each track and the next one is greater than  $IP$ .

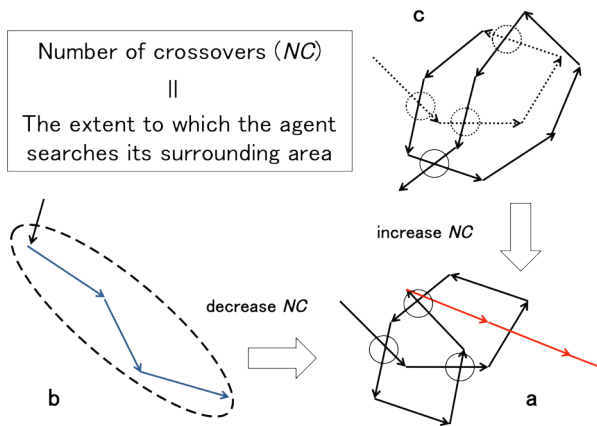


Figure 4 Schematic diagram of MEERW. (a) The MEERW model is basically same as the EERW, in which  $NC$  indicates how long an agent searches the local area. (b) A long trail stochastically generated decreases  $NC$ . (c) Excessive search time increases  $NC$ .

For the other misunderstanding, we introduce an additional memory restriction: an agent can memorize only  $N_m$  tracks as a trajectory, so it can make crossovers only with the memorized tracks (Fig. 4c). Moreover, if the agent spends  $N_s$  steps without switching phases, then  $NS$  is incremented by one because the agent misunderstands the longer local search as a rule.

In this paper, we fixed the parameters at  $N_t = 15$ ,  $IP = 0.85$ ,  $N_m = 10$  and  $N_s = 100$ .

Now, we demonstrate the Lévy-like distribution of time intervals for phase switching by comparing the EERW and MEERW models. For EERW, we measured the time steps spent in the exploitation phase as the time interval because the ratio of time steps spent in the exploitation phase to those spent in the exploration phase is constant. For MEERW, we basically measure time steps in the same way as for EERW, but if the stochastic long trail is generated, we also measure time steps from the start of the exploitation phase to the time that the trail is generated. Fig. 5 shows the frequency distribution of time intervals in a one million time-step simulation with  $SD = 0.2$  and  $P = 0.3$ . The exponent  $\mu$  is computed as the slope of a regression line for the range of values where power-law behavior (straight line in a log-log plot) is observable. The exponent  $\mu$  for EERW is 5.81 with  $R^2 = 0.957$ . For MEERW, it is 2.38 with  $R^2 = 0.943$ .

In the Lévy strategy with the Lévy intermittent model, the exponent  $\mu$  of the tail of the power-law distribution should be in the interval  $1 < \mu < 3$ . For  $1 \geq \mu$ , the distribution is not defined. For  $\mu \geq 3$ , provided the conditions of the Generalized Central Limit Theorem, the tail converges to a Gaussian distribution. In the latter case, the time interval will show an intrinsic characteristic scale. In this sense, the distribution of MEERW is considered to be Lévy-type but that of EERW is not.

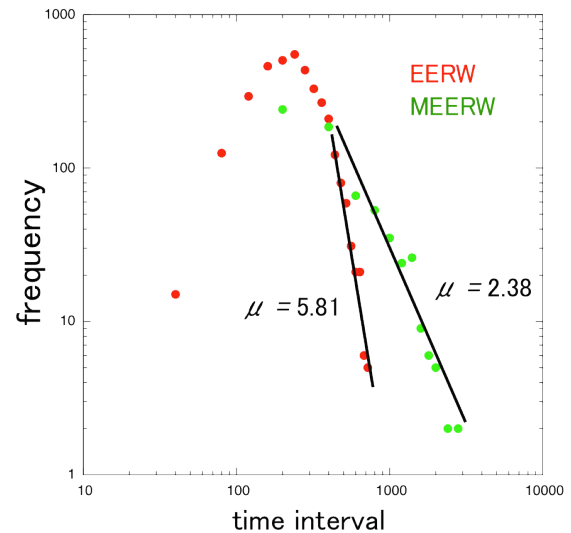


Figure 5 Power-law distributions of EERW and MEERW.

Indeed, the distribution of EERW cannot be Lévy-type. In Table 1, we estimated the exponent  $\mu$  for EERW, varying the

parameter  $NC$  from 1 to 20 and  $SD$  from 0.1 to 0.9. The results show that, for all combinations of  $NC$  and  $SD$ , the exponent  $\mu$  is greater than 3 and has a strong correlation ( $R^2 > 0.9$ ).

		$NC=1$	$NC=10$	$NC=20$
$SD=0.1$	$\mu$	5.472	5.066	4.602
	$R^2$	0.95724	0.92954	0.90619
$SD=0.5$	$\mu$	6.465	6.303	8.426
	$R^2$	0.9613	0.95194	0.93657
$SD=0.9$	$\mu$	7.771	9.071	7.691
	$R^2$	0.94909	0.95885	0.93114

Table 1 Estimation of the exponent  $\mu$  of EERW with the parameters  $NC$  and  $SD$ .

What is the difference between LW and MEERW? It is hard to distinguish the LW model from the MEERW model when we set the step length  $l$  to be very small relative to the parameter  $P$ . Fig. 6 shows a snapshot of a MEERW trajectory with  $l=0.005$  and  $P=3.0$ . It is easy to see that it has the features of an LW trajectory, such as varying step size with some small-step clusters interspersed with longer steps. The main difference between LW and MEERW lies in the behavior exhibited at the arrival point.



Figure 6 Snapshot of a MEERW trajectory with  $l=0.005$  and  $P=3.0$ .

## Discussion

The fact that local clusters are connected by saltations in animal searches and/or LW suggests that there are rules for the detection of targets (Bénichou et al., 2011). The intermittent search strategy assumes that agents move to another field if the local search is finished, which is consistent with the clustering phenomenon. This strategy implies that the agent has two different phases. However, the time interval that elapses before phase switching is given by some stochastic process, such as a Lévy process, rather than a rule.

In this paper, we started with the simple intermittent model EERW, which was not based on a Lévy process but instead was equipped with the principal features of the intermittent strategy (i.e., there were two different phases). In EERW, the

switch between phases is provided as a rule such that, if the number of crossovers exceeds a threshold  $NC$ , the agent resets the memory of trajectories and makes ballistic long trails in a direction uncorrelated with the past. We demonstrated that EERW could balance a trade-off between macro search (exploration) and micro search (exploitation), and we compared the EERW model with a CRW.

Finally, we constructed a MEERW by incorporating ambiguity or misunderstanding of the rule, in which a threshold  $NC$  is dynamically varied by the stochastically generated long trail and excessive search times. As a result, MEERW showed Lévy-like distribution. Moreover, depending on the parameter values  $l$  and  $P$ , MEERW could behave much like an LW model.

An LW model has already been constructed without a Lévy process, yet there must have been deterministic walks and interactions with a complex distribution of targets (Santos et al., 2007). Thus, our model is the first attempt to investigate the hypothesis that an LW can be generated in the absence of a Lévy process and without deterministic walks.

## References

- Bartumeus, F. and Levin, S.A. (2008) Fractal reorientation clocks: Linking animal behavior to statistical patterns of search. *PNAS*. 105(49): 19072–19077.
- Bartumeus F., da Luz M.G.E., Viswanathan G.M. and Catalan J. (2005) Animal search strategies: a quantitative random walk analysis. *Ecology*. 86(11): 3078–3087.
- Benhamou, S. (2007) How many animals really do the Lévy walk? *Ecology*. 88(8): 1962–1969.
- Bénichou, O., Loverdo, C., Moreau, M. and Voituriez, R. (2011) Intermittent search strategies. *Rev. Modern Phys.* 83: 81–129.
- Cole, B.J. (1995) Fractal time in animal behavior: The movement activity of *Drosophila*. *Anim Behav.* 50:1317–1324.
- Edwards, A.M., Phillips, R.A., Watkins, N.W., Freeman, M.P., Murphy, E.J., Afanasyev, V., Buldyrev, S., da Luz, M.G.E., Raposo, E.P., Stanley, H.E. and Viswanathan, G.M. (2007) Revisiting Lévy flight search patterns of wandering albatrosses, bumblebees and deer. *Nature* 449:1044–1048.
- Ferreira, A.S., Raposo, E.P., Viswanathan, G.M. and da Luz M.G.E. (2012) The influence of the environment on Lévy random search efficiency: Fractality and memory effects, *Physica A*. 391:3234–3246.
- Gunji, Y.P., Shirakawa, T., Niizato, T., Yamachiyo, M. and Tani, I. (2011) An adaptive and robust biological network based on the vacant-particle transportation model. *J Theor Biol.* 272:187–200.
- Humphries, N.E., Queiroz, N., Dyer, J.R.M., Pade, N.G., Musyl, M.K., Schaefer, K.M., Fuller, D.W., Brunschweiler, J.M., Doyle, T.K., Houghton, J.D.R., Hays, G.C., Jones, C.S., Noble, L.R., Wearmouth, V.J. Southall, E.J. and Sims, D.W. (2010) Environmental context explains Lévy and Brownian movement patterns of marine predators. *Nature*. 465:1066–1069.
- Humphries, N.E., Weimerskirch, H., Queiroz, N., Southall, E.J. and Sims, D.W. (2012) Foraging success of biological Lévy flights recorded in situ. *PNAS*. 109(19):7169–7174.

- Kareiva, P.M., Shigesada, N. (1983) Analyzing insect movement as a correlated random walk. *Oecologia*. 56:234–238.
- March, J.G., (1991) Exploration and exploitation in organizational learning. *Organization Science*. 2:71–87.
- MacNamara, J.M., Houston, A.I. (1987) Memory and the efficient use of information, *J. Theoret. Biol.* 125:385–395.
- O'Brien, W.J., Browman, H.I. and Evans, B.I. (1990) Search strategies of foraging animals. *Am Sci* 78:152–160.
- Reynolds, A.M. and Rhodes, C.J. (2009) The Lévy flight paradigm: random search patterns and mechanisms. *Ecology*. 90(4):877-887.
- Santos, M.C., Boyer, D., Miramontes, O., Viswanathan, G.M., Raposo, E.P., Mateos, J.L. and da Luz, M.G.E. (2007) The origin of power-law distributions in deterministic walks: the influence of landscape geometry, *Phys. Rev. E*. 75:061114-061120.
- Viswanathan, G.M., da Luz M.G.E., Raposo, E.P. and Stanley, H.E.(2011) *The Physics of Foraging: An Introduction to Random Searches and Biological Encounters*, Cambridge University Press, Cambridge
- Viswanathan, G.M., Raposo, E.P. and da Luz M.G.E. (2008) Lévy flights and superdiffusion in random search: the biological encounters context, *Phys. Life Rev.* 5:133–162.
- Viswanathan, G.M., Afanasyev, V., Buldyrev, S.V., Murphy, E.J., Prince, P.A. and Stanley, H.E. (1996) Lévy flight search patterns of wandering albatrosses, *Nature* 381:413–415.
- Viswanathan, G.M., Raposo, E.P., Bartumeus F., Catalan J, da Luz, M.G.E. (2005) Necessary criterion for distinguishing true superdiffusion from correlated random walk processes. *Phys Rev E*. 72:1– 6.

# Cooperation, Congestion and Chaos in Concurrent Computation

Mizuki Oka<sup>1</sup>, Takashi Ikegami<sup>2</sup>, Alex Woodward<sup>2</sup>, Yiqing Zhu<sup>3</sup>, Kazuhiko Kato<sup>1</sup>

<sup>1</sup>University of Tsukuba, Ibaraki 305-8577 mizuki,kato@cs.tsukuba.ac.jp

<sup>2</sup>University of Tokyo, Tokyo 153-8902, ikeg,alex@sacral.c.u-tokyo.ac.jp

<sup>3</sup>University of Tsukuba, Ibaraki 305-8577 sbbird.zhu@osss.cs.tsukuba.ac.jp

## Abstract

We are interested in understanding how conflicts for common resources can be resolved when concurrently selfish agents are in place. To answer this question, we investigate a many-core machine that performs concurrent operations. Even with the selfish and non-cooperative nature of computational processes, they successfully organize a whole task. More specifically, we use the almost lock-free (ALF) architecture, which enables effective concurrent computation on a many-core machine. A unique point of the ALF is that it performs operations on shared resources simultaneously without excluding each other. We conducted data management experiments by varying the different number of cores on a single machine and investigating the characteristic dynamics of when the highest performance is observed. We found that the temporal dynamics of the number of operations changes from noisy to bursty pattern at the optimal point. In other words, the optimal computation is found at the edge of chaos. We argue that species or agents that interact concurrently with others show chaotic behavior in a congestion state, and the cooperative state is established in the chaotic state.

## Introduction

From multi-cellular organisms to swarms of birds and a large ecological system, there is a conflict for common resources, e.g., food, territories, etc. This type of conflict can be resolved by introducing temporal oscillation. When  $N$  number of agents can periodically access the resource in turn, a happy solution can be obtained where everybody can share an equal amount of the source. This periodic behavior is realized as turn-taking behaviors (Iizuka and Ikegami, 2004; Ikegami and Iizuka, 2007). Or the conflict can be resolved spatially by each agent sticking to its own niche (i.e., food/territory) without invading space belonging to others. This spatial division of niche is often observed in ecological system and other social systems.

But what happens if agents become selfish and access the resource at a time or invade the other niche? Does it always end up with an unhappy solution where

nobody gets anything? Is it always bad manners to steal another's niche? In this paper, we investigate an artificial system that performs concurrent operations on many cores to answer these questions. More specifically, we are interested in understanding how parallel processing threads cooperatively work together and organize an entire task. We tackle this problem by having a new computational framework called "almost lock-free" (ALF for short), where we let each thread access a common work space without completely prohibiting others to access the same work space at the same time. A small interfering behavior will lead to an optimal behavior as we will show below.

ALF, presented here is a new algorithm we have invented for processing data concurrently in a computer with many cores (Wei and Kato, 2013). Due to the selfish and non-cooperative nature of computational processes, it usually is difficult to increase the throughput when performing concurrent operations on many cores. This is because, in order to maintain consistency of computation, *lock-operations* should be performed every time an operation accesses the common workspace. However, these lock operations become overhead and throughput decreases. Thus, when dealing with concurrent operations or multi-threading computation on many cores, how to process *lock-operations* are important factors to consider for achieving high throughput. ALF deals with this issue by permitting the mutual interference among processors. Using the ALF system, we will observe how concurrency causes congestions or conflicts, as well as how mutual cooperation emerges in such a system.

## Almost Lock-Free System

One of the remarkable developments of processor industry in the last decade is the serial processing speed



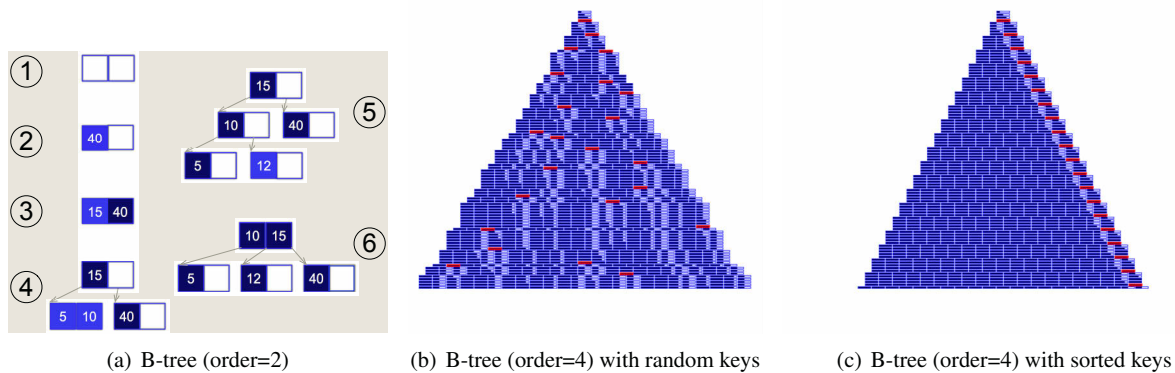


Figure 1: (a) An example of how B-tree (of order 2) is constructed. (b) An example of a large B-tree (of order 4) with randomly distributed key values. Colored space indicates the keys are inserted and uncolored space indicates the free space. Red node indicates the triggers of split. (c) An example of a large B-tree (of order 4) with ordered distributed key values.

or clock rate of core. Today, a standard computer is equipped with multi-core or even many-core processors. To make full use of these processors, concurrency control approaches have been proposed for writing concurrent programs. Dominant concurrency control approaches take what is called a pessimistic approach in which locks are performed every time a thread accesses the shared space. However, this extensive lock-based approach limits the concurrency of operations on multi-cores.

On the other hand, optimistic concurrency control approaches have been proposed. The optimistic approach assumes that multiple operations can complete without affecting each other. When conflict happens, the committed operations roll back. The optimistic approach can achieve a high throughput when conflicts are rare, since operations can complete without the expense of managing locks and without having operations wait for other operations' lock to clear. However, if conflicts happen often, the cost of restarting operations hurts performance significantly. ALF takes an approach that combines the pessimistic and optimistic concurrency control approaches, which we will explain in more detail below.

### Balanced Tree Data Structure

Many different types of file systems exist such as HFS for Mac OS, Ext for Linux, NTFS for Windows machines, ISO 9660 used on DVDs and CDs and so on. They are different in directory structure, how much spaces files are allowed to use, what sort of metadata

(about the usual data) is managed. But the basic purpose and architecture of modern file systems are similar to each other. In general, their purposes are managing access to the content of both data and the meta-data available on local and global storage devices. In particular, a data structure called *balanced Tree* (B-tree for short) is used for organizing the indices in current file systems for efficiency; B-tree supports operations such as searches, insertions, and deletions in logarithmic time efficiency. ALF uses this B-tree data structure for managing concurrent operations.

B-tree is constrained to have an equal number of pointing nodes per each node in a data-address tree. A dynamic way of constructing B-Tree under this constraint provides a unique growth of the tree form. For example, Figure 1 shows how a B-Tree grows for an input sequence of key values (40, 15, 5, 10, 12). In the example, each node can contain two values, or order of 2. A value is inserted into a node in an ascending order until the node becomes full (step 1 to 3). When the node is full, it creates two child nodes as depicted in step 4 taking the median as the parent node. Then it continues to add a value to the tree in an ascending order. It happens that the tree becomes unbalanced as in the case of step 5. When this happens, the tree adjusts itself to make it balanced by moving an adequate value to its parent node as depicted in step 6. The nodes at the bottom of the tree are called *leaf nodes* and the other nodes are called *internal nodes*.

Properties on B-tree have been extensively studied

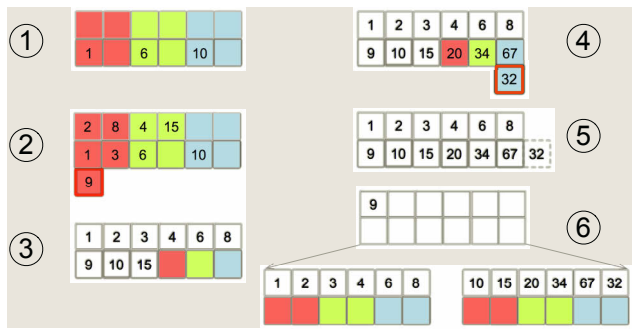


Figure 2: An example of how B-tree (of order 12) grows using ALF with three threads.

in the literature and the best-known property of B-tree is that it uses  $\log_e(2) = 69\%$  of spaces in each node when randomly distributed keys are inserted in B-tree (Johnson and Dennis, 1989). Examples of large B-trees are shown in Figure 1-(b) and -(c), with randomly distributed keys insertion and with sorted keys, respectively. These large B-trees show how leaf nodes are added or split with time. ALF uses this B-tree data structure for managing concurrent operations (i.e., insertion, deletion and searches).

### ALF on B-tree

ALF takes a hybrid approach which combines optimistic and pessimistic concurrency controls on a B-tree data structure. More precisely, ALF operates on optimistic concurrency control and only executes locks when a certain condition is met. This may cause inconsistency during the data management processes. ALF achieves this by modifying the node in the data structure and allows many threads to update the same leaf node simultaneously. A key feature to combine optimistic concept with pessimistic concurrency controls is that it gives a minimum modification for tree node structures and concurrency controls. The concept of ALF is not just about managing the data, but it also tells us how each agent should behave independently and cooperatively with a common resource.

More practically, data structure is divided into two types, *public space* and *private space*. Each core operates on private space and interacts with each other through the public space. Since operations conducted on private space are not shared on the public space, other cores cannot see even if some operations can cause conflicts on the public space. For example, an

insert operation conducted on a private space is not recognized by other cores until data is merged in the public space. If we lock the public space every time an operation affects the public space, or if we have a global clock, this conflict can be avoided.

Instead, ALF does not perform the lock every time an operation is performed on the public space, but rather lets it run until a certain condition is met, allowing some inconsistency in the data to occur. Without a global clock or complete lock operation, one may expect that the system will not self-organize anything due to the conflicts among private cores. Here we will show that this is not quite the case but rather it shows better throughput.

Figure 2 shows an example of how a B-tree grows with ALF. Here we take three threads as an example. Each thread corresponds to a core. The tree node structure is divided into two areas; the public space and the private space. The ALF adopts the idea of a thread-local area where threads can write to the private area simultaneously, to achieve a partial lock-free status. The data in the private space will be reflected on the public space by using exclusive locks when the private space becomes full. This operation is called **reorganization** and it happens when the private space for one thread becomes full. This reorganization phase is the only lock phase in the approach, and thus it is called almost lock-free.

Here, we explain how ALF works on B-tree (of order 12) by following the steps depicted in Figure 2. The initial node is assigned to private spaces and the same amount of space is allocated for each thread. Thread 1 is colored in red, thread 2 is colored in yellow and thread 3 is colored in blue, respectively. In this example, we only consider *insert* operations. We explain each step below.

- 1) Keys 1 (thread 1), 6 (thread 2), and 10 (thread 3) are inserted in each private region.
- 2) When the key 9 is assigned to thread 1 after inserting the key 2 in the same thread, it detects that the private space for thread 1 is full. This triggers reorganization of the node.
- 3) When reorganization is triggered, all the operations in the private area are reflected on the public area and keys are inserted in the public area in an ascending order. The remaining space is distributed equally for each thread.
- 4) Keys 20 (thread 1), 34 (thread 2), 67 and 32 (thread 3) are inserted on the private area. When the key 32 is inserted in thread 3, it triggers the reorganization again.

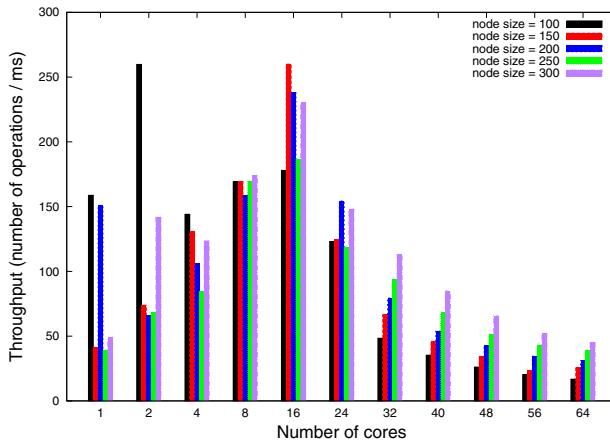


Figure 3: Throughput (number of operations executed per millisecond) when varying the number of cores and the orders.

- 5) All the keys in the private regions are inserted into the public area. However, the public area does not have enough space for all the keys in the private area. This triggers the node to split.
- 6) The medium key is taken as a parent node and two children nodes are created.

Note that private spaces are only allocated at the leaf nodes and all the internal nodes are allocated as public spaces.

### Experiments

We conducted experiments using the ALF on 64-cores machine. Here, we only use insertion as operation. The total number of keys which are randomly manipulated is 1,000,000 of the range  $[0, 1000000)$ . The order (i.e., the node size) is set to 100, 150, 200, 250 and 300. We measure the total execution time on manipulating million keys by invoking system call <sup>1</sup>, and then calculate the throughput (= number of operations per millisecond).

#### Best Degrees of “almost”?

Figure 3 shows the results of the throughput. The average throughput of the five runs is depicted in the figure. We see that the optimal throughput is obtained when the number of cores is 16 for all the orders and it decreases after that. This is because, although we gain a lot of concurrency, when the number of threads becomes larger, it also triggers a lot of reorganizations

<sup>1</sup>The system call named CLOCK\_GETTIME is used.

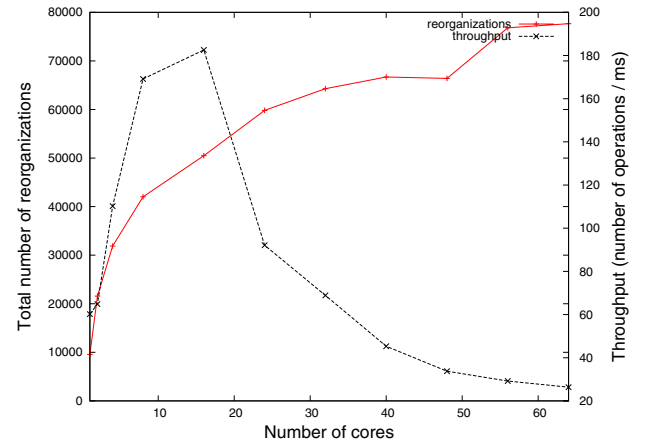


Figure 4: Total number of reorganizations and throughput with different number of cores. The throughput is maximized at 16 cores and after that the throughput decreases as the number of reorganizations increases.

and thus leads to decrease in throughput. For example, in the case of 64 cores with the order of 150, only 2 or 3 spaces are allocated for each thread; resulting in the large number of reorganizations to occur. This is confirmed in the Figure 4, which plots the number of reorganizations and the throughput in relation to the number of cores (the order is set to 150). The number of reorganizations increases as the number of cores grows.

### Characteristic Dynamics of ALF

A characteristic feature of ALF, comparing with the extensive locking system, is that a larger number of reorganizations on B-tree can occur at a time as the number of threads becomes larger. The reorganization occurs when no space exists for executing operations on any thread in the private space.

Figure 5 shows the time evolution of the internal nodes and the free space ratio at each reorganization. Characteristic dynamics is observed in the stepwise increases of the number of internal nodes. The reorganization makes new space for each thread to execute the operations and the number of free space will increase following the number of internal nodes of a tree. Having too many cores leads to frequent reorganization, slowing down the entire performance.

On the other hand, if it has enough space, many cores can work concurrently, increasing the entire perfor-

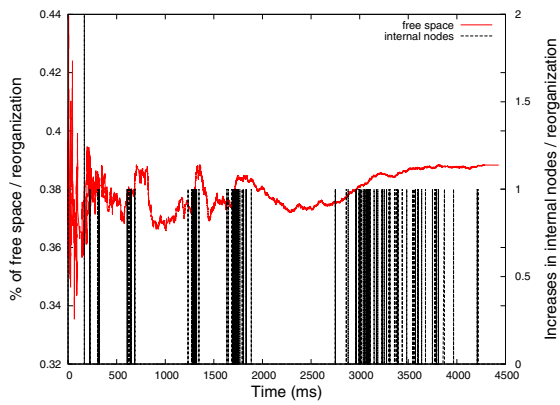
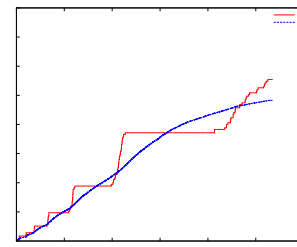


Figure 5: Dynamics of changes in the ratio of the free space in the entire B-tree and the increase in the number of internal nodes.

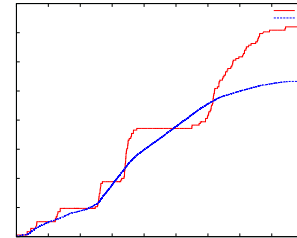
mance. As we saw in Figure 3, the maximum throughput is found around when the number of core equals to 16. If operation load is equally balanced among concurrently processing threads, the number of possible operations should be proportional to the number of cores. But actually, unbalanced operational loads occur that suppresses the number of operations resulting in the decrease of throughput.

In Figure 6, the number of leafs and internal nodes develop differently in time depending on the number of cores. For the number of cores equal to 4, the number of leafs shows a convex curve, whereas that of 64 shows the concave and that of 16 is hybrid. For the number of internal nodes, all the examples show step-wise development, except that the case of the number of cores equals to 16, the step size does not grow geometrically, but rather with some modulations. These are the pieces of circumstantial evidences that the core number equal to 16 is at the boundary of two quantitatively different dynamics phases.

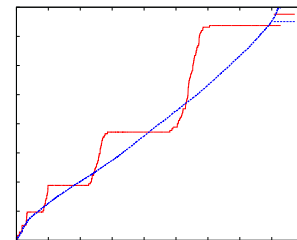
The singularity of the core = 16 is also reflected in the time evolution of the free space ratio and the number of operations. Figure 7 shows the dynamics of the number of operations and the ratio of free space of the entire B-tree. The leftmost figure corresponds to the single core case, and the right most figure corresponds to the case with 64-cores. When the number of cores is below 16, the possible number of operations over a course of time is suppressed at a lower value around 100. By further increasing the number of cores, the



(a) cores = 4



(b) cores = 16



(c) cores = 64

Figure 6: Time evolution of the total number of leaf nodes and internal nodes of the B-tree. The red colored line shows the total number of internal nodes and the blue colored line shows the total number of leaf nodes.

number of operations will be raised to around 500 with bursty time series. The critical core number 16 corresponds to the transition point. To confirm this transition, we counted the number of local peaks in the time series of the number of operations 8. We superimpose all the extracted peak values by changing the number of cores. We can observe a quantitative transition when the number of cores is 16.

We can summarize the behavior of the number of operations and of free space as follows.

- i) The number of operations can be classified into two patterns: a noisy time series with lower amplitudes and a bursty time series with larger amplitudes. The



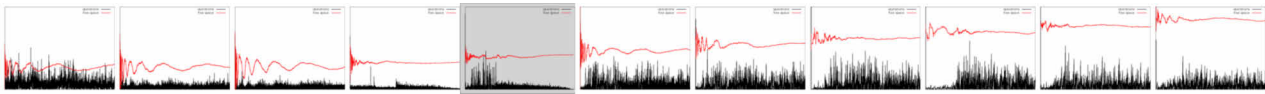


Figure 7: Dynamics of the number of operations and the ratio of free space of the entire B-tree. The figures are shown for cores = 1, 2, 4, 8, 16, 24, 32, 40, 48, 56, and 64 from left to right. The optimal throughput is found at core = 16 and is colored in grey.

optimal throughput (core = 16) is found at the transition point of these patterns. An exception is the single-core case whose time evolution of operations is similar to the optimal case.

- ii) The number of free space on average is proportional to logarithm of the number of cores.

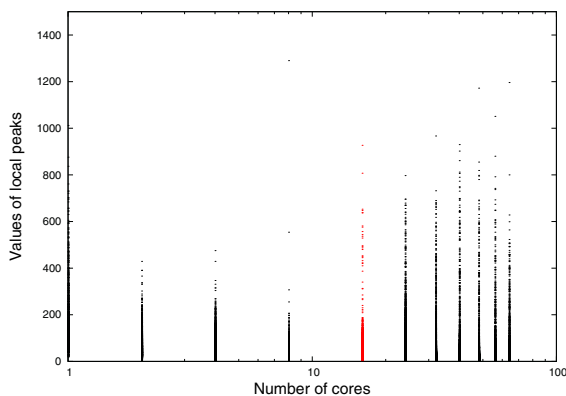


Figure 8: Counting the number of local peaks in the time series of the number of operations, we superimpose all the extracted peak values (y-axis) by changing the number of cores (x-axis). It should be noted that there is a qualitative transition at the number of cores equals to 16 (colored in red).

From these observations, we say that the optimal number of cores for the entire computation is found at the transition point, which is at the edge of the chaotic state and the bursty phase.

### Discussions

We know several examples showing that the optimal behavior can be found at the edge of chaos. This paper adds another example that the optimal computation is found at the edge of chaos and the bursty behavior. In our previous work, we have also found that the optimal

throughput of the packet switching network (PSN) at the edge of chaos and the periodic window (Ikegami et al., 2011; Takayasu, 2005). In the case of PSN, congestion of packets occurs at the critical point where the throughput becomes optimal. In an analogy with PSN, we hypothesize that congestion among different threads allows the system to perform more operations.

Concurrency causes congestion and congestion lets a system rearrange the B-tree structure, creating more free space. That is, with the increase in the size of free space, effective competition among threads is suppressed. In other words, a mutual cooperation emerges. A juxtaposition of three unrelated C-terms, concurrency, congestion and cooperation is linked by the dynamics at the edge of chaos.

A similar discussion can be applied to an ecological system's dynamics. Host and parasite networks organize a complex food web. With respect to the population dynamics of each species dynamics, we know that a weak chaos with large degrees of freedom, called *homeochaos* (Kaneko and Ikegami, 1992) leads to a network symbiotic state (i.e., cooperative phenomena). This chaotic state is attained by auto-tuning dynamics of mutation rates of each species. An initial set of species self-organizes into this homeo-chaotic state by increasing the mutation rates.

We believe that the biodiversity of a rainforest provides such an example. The abundance of each species in a rainforest is relatively low but many different species can co-exist in the same place (Connell, 1978). We argue that congestion of species produces chaotic dynamics in an ecosystem and they work concurrently. That is, species or agents that interact concurrently with others will show chaotic behavior in a congestion state, and the cooperative state is established in the chaotic state. The current work provides that the same principle can be applied in a concurrent computing system.

## Conclusion

We proposed a new idea of effective concurrent computation without using the scheme of extensively locking. A unique point of this scheme is that all the threads perform operations simultaneously without excluding each other. We found that the optimal number of cores for efficient computation is 16 in our experimental setting. The temporal dynamics of the number of operations changes from noisy to bursty pattern at the optimal point. We thus insist that the optimal computation is found at the edge of chaos. The emergence of this critical point comes from the almost lock scheme.

## References

- Connell, J. H. (1978). Diversity in tropical rain forests and coral reefs. *Science*, 199:1302–1310.
- Iizuka, H. and Ikegami, T. (2004). Adaptability and diversity in simulated turn-taking behavior. *Artificial Life*, 10:361–378.
- Ikegami, T. and Iizuka, H. (2007). Turn-taking interaction as a cooperative and co-creative process. *Infant Behavior and Development*, 30(2):278–288.
- Ikegami, T., Oka, M., and Abe, H. (2011). Autonomy of the internet: complexity of flow dynamics in a packet switching network. In *Proc. of the 20th European Conference on Artificial Life*, pages 364–371.
- Johnson, T. and Dennis, S. (1989). Utilization of b-trees with inserts, deletes and modifies. In *Proceedings of the eighth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, PODS '89, pages 235–246, New York, NY, USA. ACM.
- Kaneko, K. and Ikegami, T. (1992). Homeochaos: dynamics stability of a symbiotic network with population dynamics and evolving mutation rates. *Physica D: Nonlinear Phenomena*, 56:406–429.
- Takayasu, M. (2005). Dynamic complexity in the internet traffic. In Kocarev, L. and Vattay, G., editors, *Complex Dynamics in Communication Networks*, Understanding Complex Systems, pages 329–358. Springer Berlin Heidelberg.
- Wei, C. and Kato, K. (2013). Study on concurrent almost lock-free b-tree operations. Master thesis at the University of Tsukuba.

# Epigenetic adaptation in action selection environments with temporal dynamics

John Lones, Lola Cañamero, Matthew Lewis

Embodied Emotion, Cognition and (Inter-)Action Lab  
School of Computer Science & STRI, University of Hertfordshire  
College Lane, Hatfield, Herts, AL10 9AB, U.K.  
Johnlones@gmail.com, L.Canamero@herts.ac.uk, matt-l@semiprime.com

## Abstract

To operate in dynamic environments robots must be able to adapt their behaviour to meet the challenges that these pose while being constrained by their physical and computational limitation. In this paper we continue our study into using biologically inspired epigenetic adaptation through hormone modulation as a way to accommodate the needed flexibility in robots' behaviour, focusing on problems of temporal dynamics. We have specifically framed our study in three variants of dynamic three-resource action selection environment. The challenges posed by these environments include: moving resources, temporal and increasing unavailability of resources, and cyclic changes in type and availability of resources related to cyclic environmental changes.

## Introduction

In autonomous robotics, there is still a trend to develop and tune controllers with certain explicit goals and environments in mind (see e.g., Suganöl & Shirai, 2006; Krichmar, 2012 for an overview). This tuning can be either very direct such as pre-determining the weighting of environmental cues, or more subtle through the use of mechanisms such as reward feedback, fitness functions and activity functions (Krichmar, 2012; Lones & Cañamero, 2013).

However, even slight changes in the environment can lead to significant and often unpredictable changes in the trajectory of the same behaviour (Simon, 1969; Braitenberg, 1984; Steels, 1994; Maris & Boekhorst 1996). While environmental changes tend to modify the organism's behaviour in relation to the environmental change (see e.g., Clemens et al., 1978; Crew, 2010; Zhang & Ho, 2011), significant changes to the environment of robots possessing pre-programmed/determined adaptation mechanisms can lead to behaviours that are not only unsuitable but may render the robot inoperable (Tschacher & Dauwalder, 1999; Krichmar, 2012; Lones & Cañamero 2013).

Biological organisms are able to cope with environmental change through long-term evolutionary adaptation, more rapid ontogenetic adaptation, or through learning (Wilson et al, 1994; Cacioppo et al., 2002; Carere et al., 2005). In organisms, a form of epigenetic development occurs through

interactions with uncertain and dynamic environments (Jaenisch & Bird, 2003; Carere et al., 2005). These interactions can lead to changes in gene expression (Fowden & Forhead, 2011; Zhang & Ho, 2011) and subsequently to the appearance of new behaviours (Crews, 2011) adapted to a specific ecological niche (Narain, 2012). Recent studies have shown that hormones provide some of the signals needed to trigger the development of different aspects of the organism (Clemens et al., 1998; Crews, 2010; Fowden & Forhead, 2011).

In past experiments (Lones and Cañamero, 2013) we tested the viability of using epigenetic hormone modulation as a way to allow a robot to adapt to unknown environments. In that study, we placed the same architecture into various environments posing different challenges to the robot. For each experiment, we researched the ability of the epigenetic robot to develop unique behaviours in direct relation to the environmental challenges. In all cases, a significant increase in viability was noticed in the epigenetic model compared to an architecture lacking the epigenetic mechanism.

In the present study, we investigate the ability of a robot, endowed with the same architecture as in the above-mentioned study, to cope with environments posing different types of temporal dynamics problems. In our previous study, the environment we used, while possessing some dynamic qualities, were predominantly static. Changes in that environment occurred as a consequence of the robot's actions. However, in this study, each environment has its own dynamics. This creates an opportunity to examine the robot's behaviour when faced with constantly changing and potentially unpredictable environments.

## Robotics model

**The robot** we have used in this study is the Koala II ([www.k-team.com](http://www.k-team.com)), a medium-sized wheeled robot. It is equipped with 16 infrared (IR) sensors placed around its body, and we use them as both proximity sensors and touch sensors. Proximity IR sensors are grouped to monitor the eight cardinal and ordinal directions surrounding the robot. In our case, this permits the detection of the direction that possesses the least resistance to movement. Touch IR sensors "extend" the

robot's body by 1cm—what we refer to as “extended body”. Any encroachment of this area is categorised as contact, and the force of contact is dependent upon both velocity and persistence of the encroachment. Finally, we have fitted a webcam to the robot that, in combination with OpenCV, allows the robot to track specific coloured objects. For a more detailed overview of our setup please see (Lones and Cañamero, 2013).

**The Physiology** of the robot consists of three survival-related homeostatic variables, which must be maintained within a pre-set boundary ( $0 \leq h.var_i \leq 100$ ) for continued survival (see table 1). These three survival-related homeostatic variables are based upon plausible robotic needs in form of energy (E), physical condition (C) and temperature (T).

The robot's *energy* depletes at a rate equivalent to a basal metabolic rate plus the energy cost of activating subsystems such as vision. Since these subsystems are always active in this implementation, energy decreases at a constant rate of  $\delta$  per step. *Condition* represents a measure of health for the robot. Deficits occur in a semi-unpredictable manner from collisions. Both variables can be recovered by finding and consuming specific resources. Finally, *temperature* represents the internal heat level of the robot. The robot's temperature rises as a function of a combination of the environment's ambient temperature and the robot's movement speed. Cooling down (dissipation of temperature) occurs at a constant rate. Assuming a moderate or rapid dissipation of excess heat, the robot is able to maintain a steady speed without running the risk of overheating. Table 1 provides an overview of the different internal variables.

H.Var	Ideal Value	Limit	Cause of deficit	Recover per step
E	100	0	$\delta = 0.1 \downarrow$	$1 \uparrow^*$
C	100	0	Contact $\downarrow$	$1 \uparrow^*$
T	0	100	Movement $\uparrow$	$0.8 \downarrow$

\* The robot must be near the resource for recovery to commence

Table 1: The homeostatic variables of the robot. In this implementation, if Energy or Condition fall below 0, the robot “dies”. Temperature has an inverse effect.

These survival-related homeostatic variables give rise to a Viability zone (Physiological space), following Ashby (1952) and Avila-García and Cañamero (2004). The position in and management of the dynamics of the viability zone provide different ways to quantitatively measure the robot's performance and wellbeing. Like Avila-García and Cañamero (2003, 2004) and our earlier paper Lones and Cañamero (2013) we have used this idea of the viability zone to create a performance indicator called “comfort”. Comfort provides a measure of the average homeostatic deficit at any time, and the “risk of death”, which indicates how close the internal state is from reaching lethal values. Comfort is calculated on a scale of 0 to 1; with a comfort level close 1 indicating homeostatic variables near their ideal levels. Whereas a comfort level near to 0 would indicate large homeostatic deficits and a high “risk of death”. Along with the comfort

level the standard deviation at specific points is also provided. This allows for a greater insight into the robot's performance.

## Hormones

Apart from providing a measure of wellbeing, the tendency to satisfy homeostatic needs provides part of the foundation for the formulation of motivations. Internal needs modelled as homeostatic variables have long been used to model motivations in robotics, providing efficient and understandable simple models that permit the generation of appropriate goal-oriented movements and behaviours (e.g., Cañamero, 1997; Breazeal & Scassellati, 1999; Arkin, 2003; Bach, 2011). However, in biological systems matters are more complex, as motivations do not come directly from homeostatic deficits. Rather, hormone secretion derived from homeostatic deficits (e.g., ghrelin in the case of hunger) are shown to be behind the formation of motivation (Wallen, 2001; Malik et al., 2008) and the motivational value of environmental cues (Wied, 1976; Martinez, 1981; Frijda 1986). The development of an organism's hormonal gland activity (in the form of synthesis and release) as well as the development of receptor sensitivity are believed to be susceptible to both endogenous and exogenous environmental cues (Zhang & Ho, 2011). This would suggest that motivation is also in part affected by past experience.

An epigenetic hormonal motivation-like system could potentially provide an efficient method to allow robots to align their needs and goals with challenging environments on a more permanent basis, e.g., to “grow up” adapted to an environment presenting uneven opportunities to fulfil survival-related needs. This process would affect the tolerance to different homeostatic deficits and the priority with which they would be maintained as a function of the developmental environment. Through such an epigenetic process, during the earlier stages of the development of the robot, its hormone glands associated with underrepresented needs would become more sensitive. That is, smaller homeostatic deficits would trigger the same level of hormone secretion as we would see in robots that had “grown up” in a more balanced environment.

Hormones are however not limited to motivations. In our previous study (Lones and Cañamero 2013), we showed how an epigenetic hormone-like system can give rise to diverse behaviours tailored to different environments. While hormone-modulated behaviours had already been successfully modelled in the past, what sets our model apart from others such as (Avila-García and Cañamero, 2004; and Krichmar, 2012), is that: (a) instead of having a limited number of pre-set behaviours, behaviours emerge from the combination of the hormone-activated sub-systems within the robot; and (b) that due to the epigenetic nature of the hormone glands, this means that two robots with the same motivational tendency but with different developmental histories may behave in different ways.

## The Action selection mechanism

The ASM incorporated a “voting-based” (VB) policy based upon ideas presented by Tyrell (1993). By using the VB architecture, actions selected by the robot will be those that



provide the greatest overall benefit. In comparison, a “winner-takes-all” (WTA) policy would lead to the selection of the actions that satisfy the current greatest need. Although Avila-García et al. (2003) found that a WTA outperformed the VB architectures in dynamic environments, in their environments the dynamics was introduced by the presence of predators, thus posing very different challenges. Using our model, in preliminary experiments we found that the VB architecture performed better, as shown in figure 1. These preliminary experiments consisted of five 5-minute runs of each architecture type. Performance was measured using comfort as an indication of the robot’s wellbeing.

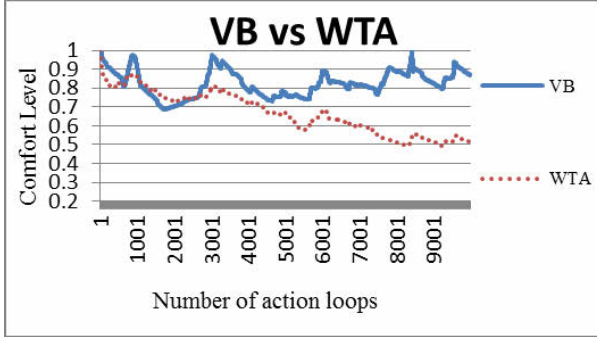


Figure 1: Comparison of the performance of VB and WTA architectures.

### Hormone System

At the core of the VB ASM lies a hormone-like system influenced by models developed by Avila-García & Cañamero (2004) and Krichmar (2012). In a new development, we have implemented two different types of hormone, which are classed as either endocrine hormones (Eh) or neurohormones (Nh) (see Table 2). Drawing on biological systems, our Eh-like implementation consists of hormones with the primary purpose to try to maintain homeostasis (Murphy & Bloom 2006). The Eh group is made up of three hormones: one associated with each homeostatic variable. For each hormone,  $h$ , secretion occurs via a gland,  $g_h$ , and the rate of secretion,  $s_h$ , depends upon the current homeostatic deficit,  $d_h$ , and the activity level of the gland,  $\Psi_h$ ,

$$s_h = k(\Psi_h^{d_h}) \quad (1)$$

where  $k$  is a constant that scales the size of secretion. Once released, each secretion persists in the system for a random number of action loops (within a fixed range) before decay of that particular secretion occurs. The larger the secretion, the longer it will take to fully decay. The concentration ( $conc_h$ ) of these hormones is thus determined by the total sum of each active secretion.

The second group of hormones, Nh, contains only one hormone, D1. This hormone facilitates what can be described as “dominant” or potential “aggressive” behaviour. This is achieved by having the hormone suppresses environmental cues that are associated with negative stimuli. For example a robot with a high D1 level that detects a desired resource will move towards it directly at a high speed pushing aside any obstacles, disregarding the potential of damage from

collisions. In contrast a robot in the same situation but with a low D1 level would instead move around obstacles to reach the desired location.

Rather than being triggered by internal deficits, as with Eh hormones, Nh secretion is linked to the mean of the external environmental cues ( $ec_{dt}$  where  $d$  is the direction of the cue and  $t$  the type e.g. energy or repair source). Therefore the  $conc_h$  of the neurohormone is determined by

$$conc_h = (conc_h\beta) + \alpha\bar{ec}\Psi_h \quad (2)$$

where  $\alpha$  is a predetermined weighting factor ( $0 \leq \alpha \leq 1$ ) and  $\beta$  is the disperses rate of the hormone which is set to 0.9 (leading to a 10% disperse rate each loop) during these experiments.

Also different to the Eh model,  $\Psi_h$  is not a set value. Instead, the activity level of the gland is stimulated by the mean concentration of the Eh hormones ( $conc_{Eh}$ ) in the body, similar to tropic hormones. Where in biological systems, these hormones have been demonstrated to cause/increase the secretion/production of other hormones (Sherwood, 2003):

$$\Psi_h = e^{-4e^{-0.5conc_{Eh}}} \quad (3)$$

$\Psi_{D1}$  will lie between a value of 0 and 1, with a value of 1 indicating the gland is fully active, and 0 signifying that the gland is inactive. The final part of the Nh equation models neuroreceptor sensitivity ( $sen_h$ ).

$$streng_h = e^{de \times e^{sen_h conc_h}} \quad (4)$$

where  $streng_h$  the cumulative effect of the neurohormone on the system once the concentration and sensitivity to it are taken into account,  $de$  is the minimum stimulation needed for activation of the receptor, and  $sen_h$  the sensitivity of the receptor to the hormone.

H. Name	H. Type	Trigger	$\Psi_h$
E1	Eh	Energy deficit	0.09
C1	Eh	Cond deficit	
T1	Eh	Temp deficit	
D1	Nh	Visual cues	Varies

Table 2: Robotic hormones

### Hormones and the ASM

The VB ASM consists of a two-step computation (see figure 2). The first step calculates the current homeostatic motivations or drives ( $m_{dt}$ ) (see Table 3). Although three drives are present, we only need to directly calculate the intensity of hunger and damage. The hyperthermia drive, which can be satisfied by reduced or no movement, instead suppresses other drives. In addition to the internal state, motivations are influenced by environmental cues,

$$m_{dt} = \begin{cases} \frac{conc_h(t)ec_{dt}}{1+conc_{T1}} & \text{if } strength_{D1} > 0.75 \times conc_{T1} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

The perceived environmental cue in the forward direction,  $ec_{2t}$ , is given an additional +1 score to simulate a restless mechanism and allow forward movement without external stimuli. To further reduce excessive switching of motivations  $ec_{2t}$  is given a 10% bonus to its value as a form of “hysteresis”.

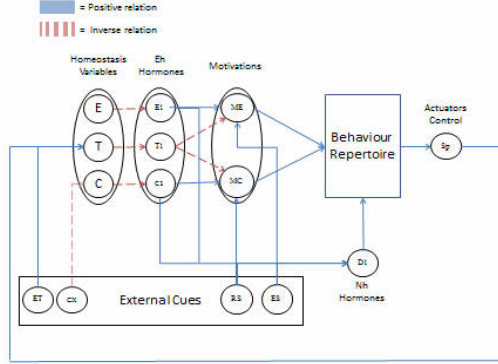


Figure 2: Hormone-based Architecture.

Motivation	Drive	ext Stimuli	Suppressed by
Hunger (ME)	E1	Eng source	T1
Damage (MC)	C1	Rep source	T1
Hyperthermia	T1	Climate	E1 & C1

Table 3: Motivations of the robot

The second step in the ASM calculates the behaviour to execute given the current motivational state and environmental conditions. Unlike previous hormone-based architectures such as Avila-García & Cañamero (2004) and Krichmar (2012) no explicit behaviours have been modelled. In our case, behaviours occur from dynamic combinations of different systems with no pre-set physiological cost or gain. The cost or gain of behaviour execution results from the sum of physiological changes that occurred during the action.

One of these subsystems is the robot’s personal space (Ps) (see Hall, 1966), an area that the robot will treat almost as an extension of its own body. Using a similar technique as with the “extended body” (the IR-based touch sensors around the robot’s body), the robot will normally maintain the Ps free from other objects. The radius of the Ps zone is determined by the current C1 hormone concentration. Encroachment will lead to attempts to re-establish a space by moving along the path of least resistance ( $d_n$ ), with a slight preference to going forward. D1 counteracts the tendency to keep the Ps empty, allowing objects within the Ps while trying to satiate drives. At high levels D1 will facilitate physical contact, allowing the robot to push or “attack” anything standing between itself and its target, the size of the Ps at any given time is show in equation 6.

$$Ps = \left( ns + ns \frac{conc_{C1}}{conc_{C1\{max\}}} \right) (1 - strength_{D1}) \quad (6)$$

where  $ns$  is the normal, unadjusted size of personal space, and  $conc_{C1\{max\}}$  the maximum potential concentration of the hormone.

## Hormone-Signalled Epigenetics

The final and following aspect of the model introduces an epigenetic adaptation mechanism into the architecture. Taking inspiration from recent biological studies (see Crews, 2008 & 2010, Fowden & Forhead, 2011 for an overview) hormones trigger epigenetic changes in the robot. In our robot, hormone levels both indirectly and directly provides a fairly accurate measure of current conditions in the environment and level of situatedness. For instance, the current level of the E1 hormone is an indication of how well the robot is managing its need for energy. Combined with the concentration of D1, it is possible to determine the root of the imbalance, as either issue of scarcity, or difficulty of access to the resources.

These hormones can thus act as signals for epigenetic adaptation, whereby development of the glands that secrete hormones and receptors that receive them are influenced by the external environment. For example, an autonomous robot that is often low on condition/health will have a high concentration of the C1 hormone within its system. The high concentration will lead to a long-term increase in the activity level ( $\Psi$ ) of the gland that secretes C1. This will mean that sub systems such as the desire to maintain a degree of personal space or find repair resources will be much more prevalent within the model. Formula 7 shows method used to facilitate the epigenetic change in activity levels ( $\Psi$ ) of the gland for hormones in the eh group.

$$\Psi_h += \frac{conc_h}{l} \quad (7)$$

where  $l$  is a constant to regulate the speed of epigenetic change.

Formula 8 shows the method in which epigenetic change can occur to the sensitivity of neurohormone receptors for the hormones in the Nh group

$$sen_h -= \frac{(strength_h - 0.05)}{j} \quad (8)$$

where  $j$  is a constant to regulate the speed of epigenetic change.

Drawing on the notion of critical periods in biological organisms, the epigenetic process above is active during the early period of the robot’s life. This critical period represents a window frame when organisms are most susceptible to the influences of external perturbations (Winks & Berthouze, 2008), mediated via hormone modulation (Crews, 2010), among other things.

## The temporal three-resource problem

The architecture described here has been tested in a temporal three-resource action selection problem framework, in which a robot needs to timely and appropriately select among and satisfy three needs using resources available in the environment in order to survive (remain operational or “alive”). Our experimental design included three different sets of experiments corresponding to three variants of an environment that pose different challenges arising from the temporal dynamics of the resources. Each set took place within a 2mx2m bordered environment inhabited by a single robot. Within each environment a number of energy and repair resources were available to allow the robot to replenish homeostatic deficits. These resources were represented by two different coloured sets of balls. The environments also contain an ambient temperature that is sensed internally by the robot.

**Scenario one** consists of the base environment with one of each resource moving in a continuous pattern at a constant speed, slightly faster than the robot’s average, around the arena, see figure 3. At the end of each movement path (represented by a letter) the resource would pause for a period of 2 seconds. In cases where the robot was in the direct path of a resource, the resource would be manoeuvred around the robot using the shortest path before returning to its original trajectory. In the case where the resource was pinned or the movement was blocked by the robot, no attempts were made to push the robot aside. Instead, movement of the resource was halted until the robot moved away and a viable path was visible. At the start of each run the resources started at a different opposite points, e.g. A and E.

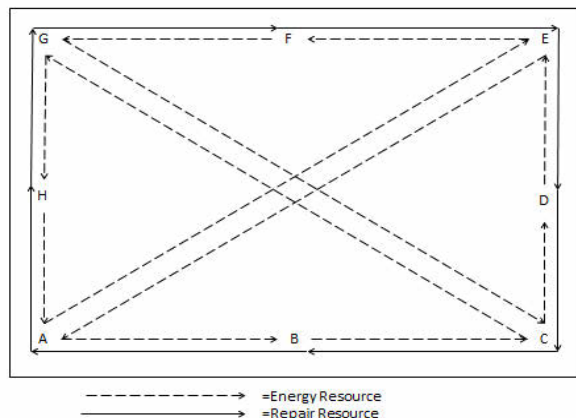


Figure 3: The pathways and start points of the resources.

**Scenario two** was again based on the base environment. However, in this scenario the energy resource appears at set points within the environment once every minute, the period during which it is available reduces over time, i.e., it becomes decreasingly available. For the first five runs, the energy source would remain for 30 seconds before being removed. In the second five sets the duration was reduced to 20 seconds and the final five saw the resource only accessible for 10 seconds of every minute. The set points are the same as the start of pathways as seen in figure 3. In order to avoid biases, the order of set points where the resource would appear was

predetermined randomly before each run. The choice to have the temporal properties apply to only the energy source was done to examine the robot’s ability to deal with the increasing disparities between the availability of the repair and energy sources.

It is worth noting that the robot has no capacity to monitor time. Therefore, there is no facility to try to directly predict when the resource will appear. Rather, over time the robot will adapt to the scarcity and rarity of the resource. The use of a strict time period was to ensure each robot had same constraints and opportunities.

**Scenario three** examines the ability of the robot to adapt to the effects of dynamic climatic changes. In this experiment the standard base set up of the environment was used with one of each resource available at all times. However the ambient temperature of the environment would increase and decrease over time, simulating a day-and-night temperature cycle. The entire cycle lasts for four minutes, as can be seen in figure 4. To simplify the model, ambient temperature ranked between 0 (cold) and 10 (scorching heat).

In order to increase the dynamics of the environment, temperature was allowed to fluctuate by up 2 points to simulate potential meteorological phenomena. The fluctuations were calculated at start of each 10-second period and lasted until the next period.

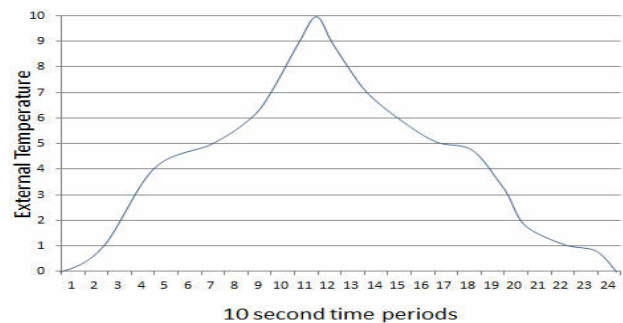


Figure 4: An example of an average weather cycle with meteorological phenomena. The periods between 6 and 18 or minute 2 and 3 are analogue to daytime, with the highest temperature occurring midday equivalent to the sun at its peak in a natural environment.

## Experiments and Results

The robot was tested over a total of 35 runs split in 10/15/10 runs amongst the three previously described scenarios. Each run lasted a maximum of 10,000 steps around 10 minutes 40 seconds per run. The epigenetic system was active during the first 3 minutes (2880 steps). A second set of runs was conducted in the same manner for a robot without the epigenetic mechanism to serve as a basis for comparison. The viability of both architectures was assessed using the previously discussed *Comfort* measure and standard deviation as well as visible observation. In cases where a robot died before the end of a run, a comfort value of 0 would be recorded for any remaining loops.

### Scenario One

This environment provides the robot with two distinct challenges. The first and most obvious was the need to develop a consumption behaviour suitable for moving resources. Secondly, this environment presents the first situation where the robot can be damaged by other elements (objects or organisms) of the environment. While as previously stated resources will move around the robot if it is directly in their path, they will still move close enough to encroach upon the extended body, causing damage. Therefore, the robot will also need to adapt to co-exist with the resources, not just how to exploit them. The results of the first experiments can be seen below in figure 5.

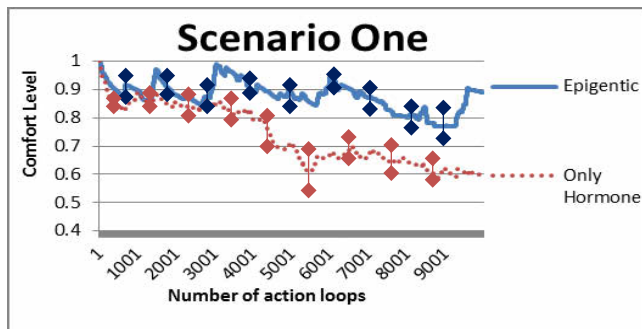


Figure 5: The combined results for Scenario One.

As can be seen in figure 5, the epigenetic robot performed at a higher level overall, but more interestingly had a much lower standard deviation of 0.05 compared to 0.17. The differences in standard deviation can be attributed to the dynamic nature of the resources. In some situations the robot was positioned in the ideal location to catch and consume resources as they passed. This led to timely management of the robot's homeostatic needs. However, in other cases the robot would need to actively move across the arena and chase a resource. Since the resources moved slightly faster than the robot's average speed, the motivation to consume the resource had to outweigh the motivation to limit speed in order maintain a low temperature.

Distinctive behaviour developed for each of the architectures in this environment. The epigenetic model would develop an "ambush-like strategy": the robot would remain sedentary until an energy source passed closely, at which point the robot would give chase at full speed often pinning the resource to a wall until it had finished consuming it.

In contrast, the non-epigenetic model would engage in "drawn-out chases". As the motivation to consume the resource allowed it to generate the speed needed to catch up, excess heat was generated. This heat generation led to premature end of the chase on a number of occasions. Finally the epigenetic robots displayed more adaptive behaviour at avoiding unnecessary collisions with resources, and almost no unwanted collisions occurred after the early periods.

### Scenario Two

In scenario two, we tested the ability of the robot to deal with resources only available for limited periods of time. The 15

runs were divided into 3 groups of increasingly challenging runs with the resource present for 30/20/10 seconds of every minute, challenging the robot to act in a timely manner when the opportunity to recover from homeostatic deficits was present. This temporal quality only applied to the energy resource. This further challenged the robot to overcome the "distraction" of the more readily available repair resource. The results of this scenario can be seen below in figure 6.

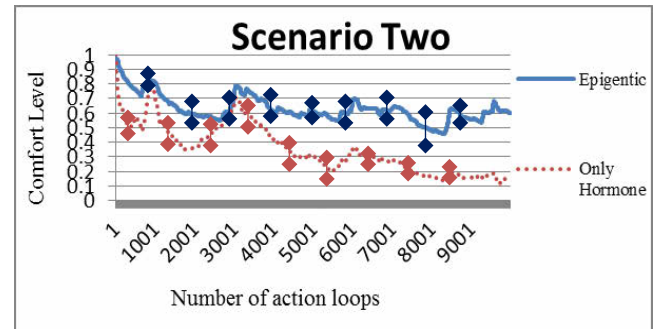


Figure 6: The combined results for Scenario Two.

Both robots performed at a similar level during the first five runs with 30 second window of opportunity. While the epigenetic robot moved more promptly to resources when they appeared, neither robot ever was in any real danger. However, as the window of opportunity shrunk, the differences between the two models became very apparent, as can be seen in figure 6.

As the point where the resource would appear next was unknown to the robot, it was inevitable that both architectures would miss some opportunities to replenish. However, the epigenetic model was generally quicker to find any resource due to the development of the E1 and D1 glands, thus giving the robot a greater chance of survival also when the opportunities were missed.

Finally, due to missed opportunities to fully recover deficits, both robots often contained significant level of the D1 hormone. This in turn resulted in higher occurrences of collision in later runs, subsequently increasing the need for repair resources. In multiple cases this led to similar levels of need for both the energy and repair resource. This resulted in the non-epigenetic robot sometimes going to the readily available repair source during the limited periods when the energy source was present and seen. This occurred on some occasions even when condition deficits were not significant. In contrast, the epigenetic model had adapted to the rarity of the resource. It only missed the opportunity to replenish energy once. This occurred when its condition levels were critical. In total, 7 of the non-epigenetic robot runs ended prematurely compared to a single death in the epigenetic model. Due to the high level of fatalities, the hormone-only model actually had a lower standard deviation of 0.03 in contrast to 0.08 in the epigenetic model.

### Scenario three

In the final scenario we tested the ability of the two robot architectures to deal with cyclical climates, with the cycle of change in ambient temperature previously shown in figure 4. Like scenario two, this environment challenged the robot's



ability to take advantage of limited windows of opportunity. During the periods where ambient temperature reached its peak, even limited movement soon led to overheating. Two of each of the resources, spread evenly in each corner, were constantly available in the environment. The results for this experiment can be seen in figure 7.

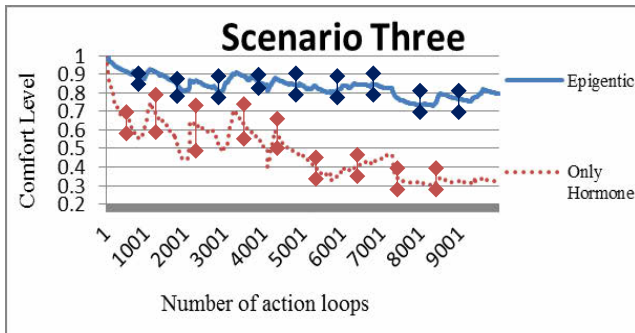


Figure 7: The combined results for Scenario Three.

As can be seen, the epigenetic robot had much greater success. After the initial 3 or 4 cycles the robot's hormone glands had developed in such a way that, during periods with the highest ambient temperature, virtually all actions would be suspended. As soon as the ambient temperature dropped, the robot would move to replenish any deficits. The epigenetic robots developed two contrasting behaviours in order to survive the periods of high ambient temperature. One group simply over consumed and in effect "hibernated". The second group would instead stay near the energy source at all times apart from the occasional need to repair, allowing itself to consumer energy during the increased climate with only very limited movement needed.

In contrast, the non-epigenetic model often ran low on energy during the day cycle. This resulted in the robot being forced to move to energy sources, generating significant overheating, which led to the death of the robot on 3 occasions.

## Conclusion

In our past study (Lones and Cañamero 2013) we have shown how epigenetic changes through hormone modulation increase the adaptability of a robot. Specifically we demonstrated how this process leads to behaviours tailored to specific environmental niche. These robots were placed into different environments with exactly the same starting architecture. However, through epigenetic processes, the robots developed distinct traits and behaviours depending on the environment in which they developed.

In the study presented in this paper, we have investigated the same architecture under new criteria. Specifically, we focused on the ability of the robot to adapt to environments that presented temporal dynamics challenges. In the first experiment, the robot needed to adapt to fast-moving resources. While the robot could simply have "chased after" the resource at top speed, this would lead to unwanted overheating and would not guarantee appropriate satisfaction of its homeostatic needs. Instead, the robot developed what could be considered equivalent to an "ambush-like hunting

tactic". In the second experiment, the robot was challenged to adapt to limited windows of opportunity to satisfy a homeostatic need, all the while needing to adapt and disregard opportunities offered by more easily available resources that permitted to satisfy other needs. Needing to find a balance between maintaining the different homeostatic needs, the robot was able to respond in a timely manner to rare occurrences while still finding time to satisfy the other needs. In the final experiment, we examined the ability of the robot to adapt to cyclical events. Under this scenario, the robot needed to fully utilise the cooler periods of the day, which allowed it to be in a position to survive hotter periods when most actions would need to be suspended. This experiment marked the first time we saw the epigenetic model divide into two distinct groups. Each group developed a different method to deal with the debilitating temperature.

As we have shown, epigenetic adaption though hormone modulation potentially offers a suitable method to allow a base architecture to develop behaviours to adapt to environments presenting different temporal challenges.

## Acknowledgements

John Lones is supported by a research studentship of the University of Hertfordshire. Partial support was also provided by the FP7 project ALIZ-E. We would like to thank the other members of our lab for their feedback. John is also grateful for additional feedback and advice from Zornitsa Trendafilova.

## References

- Arkin, R. (2003). Moving up the food chain: motivation and emotion in behavior based robots. In *Who Needs Emotions: The Brain Meets the Robot*, Fellous J., Arbib M., editors. Oxford University Press, p.p 245–270.
- Ashby, W. R. (1952). *Design for a Brain*, Chapman and Hall, London
- Avila-García, O., Cañamero, L. and Boeckhorst, R. (2003). Analyzing the performance of 'Winner-Take-All' and 'Voting-Based' action selection policies within the two-resource-problem. In *Proceedings of seventh European conference in artificial life (ECAL03)* p.p 733–742
- Avila-García, O. and Cañamero, L. (2004). Using hormonal feedback to modulate action selection in a competitive scenario. In: *From Animals to Animats: Proceedings of the 8th International Conference of Adaptive Behavior (SAB'04)*, Cambridge, MA, MIT Press, p.p 243–252.
- Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology* Cambridge, MA:MIT press/Bradford Book
- Breazeal, C. and Scassellati, B. (1999). "A context-dependent attention system for a social robot," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI 99)*. Stockholm, Sweden, 1146–1151.
- Cacioppo, J., Brenston, G., Taylor, S. and Schacter, D. (2002). *Foundation in Social Neuroscience* Cambridge, MA:MIT press
- Cañamero, L. (1997). Modeling motivation and emotions as a basis for intelligent behaviour. In W.I Johnson editor *proceeding of the first int conf on autonomous agents (agent 97)*, pages 148-155. New York: ACM Press
- Carere, C., Drent, P., Koolhaas J. and Groothuis, T. (2005). Epigenetic effects on personality traits: early food provisioning and sibling competition *Behaviour* volume 142 number 9-10, p.p 1329-1355
- Clemens, L., Gladue, B. and Coniglio, L. (1978). Prenatal endogenous androgenic influences on masculine sexual behaviour and genital morphology in male and female rats. *Hormones and Behavior* volume 10 issue 1, p.p 40-53

- Crews, D. (2008). Epigenetics and its implications for behavioural neuroendocrinology. *Frontiers in Neuroendocrinology* 29 p.p 344-357
- Crews, D. (2010) Epigenetics, brain, behavior, and the environment *HORMONES*, 9(1): p.g 41-50
- Fowden, A. L. and Forhead, A. J. (2009). Hormones as epigenetic signal in developmental programming, *Exp Physiol.* June; p.p 607-25.
- Frijda, N. (1986). *The Emotions*, Cambridge University Press
- Jaenisch, R. and Bird, A. (2003). Epigenetic regulation of gene expression: how the genome integrates intrinsic and environmental signals. *Nat Genet* 33(Suppl): p.p 245–254
- Krichmar, J. L. (2012). “A biologically inspired action selection algorithm based on principles of neuromodulation” in *The 2012 International Joint Conference on neural networks* p.p 1-8
- Lones, J. and Cañamero, L. (2013). Epigenetic adaptation through hormone modulation in autonomous robots. Submitted for publication
- Malik, S., McGlone, F., Bedrossian, D. and Dagher, A. (2008). Ghrelin modulates brain activity in areas that control appetitive behavior. *Cell Metab* 7:400–409
- Maris, M. and Te Boekhorst, R. (1996). Exploiting physical constraints: heap formation through behavioral error in a group of robots. In *Proc. IROS'96, IEEE/RSJ International Conference on Intelligent* p.p 1655-1660
- Martinez, J. L. (1981). *Endogenous Peptides and Learning and Memory Processes* Academic Press Inc
- Murphy, K. G. and Bloom, S. R. (2006). Gut hormones and the regulation of energy homeostasis. *Nature* 444 p.p 854-859
- Narain, C. (2012). Changing behaviour with epigenetics, *Natural Neurscience* 15, p.p 1329
- Sherwood, L. (2003). *Human Physiology: From Cells to Systems* Wadsworth Publishing Co Inc; 5th Revised edition
- Simon, H. A. (1969). *The Sciences of the artificial*. Cambridge, Mass.: MIT Press.
- Steels, L. (1994). A case study in the behaviour oriented design of autonomous agents
- Sugano, S. and Shirai, Y. (2006). Robot design and environment design Waseda Robot House Project”. In *Proc. SICE-ICASE* 31–34. Busan
- Tschacher, W. and Dauwalder, J. P. (1999). Dynamics, synergetics, autonomous agents—nonlinear systems approaches to cognitive psychology and cognitive science World Scientific Publishing Co Pte Ltd
- Tyrrell, T. (1993). *Computational Mechanisms for Action Selection* PhD thesis University of Edinburgh
- Wallen, K. (2001). Sex and context: hormones and primate sexual motivation. *Horm. Behav.* 40, 339–357
- Wied, D. (1976). Hormonal influences on motivation, learning, and memory processes. *Hosp Practice* 11:123–131
- Sloan Wilson, D., Clark, A. B., Coleman, K. and Dearstyne, T. (1994). Shyness and boldness in humans and other animals *Trends Ecol Evol* 9 p.p 442-446
- Winks, O. and Berthouze, L. (2008). Predicting the timing of critical periods in development In *Proceedings of the 8th International Conference on Epigenetic Robotics*, pp. 127-134
- Zhang, X. and Ho, S. M. (2011). Epigenetics meets endocrinology *Journal of Molecular Endocrinology* 46, p.p 11–32

# Hidden information transfer in an autonomous swinging robot

James Thorniley<sup>1</sup> and Phil Husbands<sup>1</sup>

<sup>1</sup>Centre for Computational Neuroscience and Robotics, Dept. of Informatics, University of Sussex, UK, BN1 9QG  
jt241@sussex.ac.uk

## Abstract

This paper describes a hitherto overlooked aspect of the information dynamics of embodied agents, which can be thought of as hidden information transfer. This phenomenon is demonstrated in a minimal model of an autonomous agent. While it is well known that information transfer is generally low between closely synchronised systems, here we show how it is possible that such close synchronisation may serve to “carry” signals between physically separated endpoints. This creates seemingly paradoxical situations where transmitted information is not visible at some intermediate point in a network, yet can be seen later after further processing. We discuss how this relates to existing theories relating information transfer to agent behaviour, and the possible explanation by analogy to communication systems.

## Introduction

The dynamics of embodied agent-environment systems are increasingly analysed using information theory (Lungarella and Sporns, 2006; Pfeifer et al., 2007b; Bertschinger et al., 2008; Klyubin et al., 2008; Pitti et al., 2009; Williams and Beer, 2010; Moiola et al., 2012; Schmidt et al., 2012). This paper adopts this approach and demonstrates a phenomenon that is consistent with the analogy to communications, but thus far seemingly overlooked in studies of information transfer in embodied agents. We describe “hidden” information transfer in a simulated robot: strongly physically coupled parts of the system carry information between separated endpoints, without such information transfer being visible between the carrier components themselves.

Information transfer is often characterised using forms of *transfer entropy* (Schreiber, 2000), itself a nonlinear generalisation of Granger causality (Barnett, 2009). Information transfer from  $X$  to  $Y$  is quantified by the relative improvement in statistical prediction of the future states of  $Y$  when the current state of  $X$  is known in addition to the already-known historical states of the target variable  $Y$ . A known issue, and potential source of confusion, is that information transfer is not a measure of the *physical strength* of coupling – a common example being synchronised systems, where very high coupling may mean that two time series are almost

identical, leading to little prediction improvement and hence low transfer entropy in spite of strong physical coupling (this is seen in e.g. Thorniley, 2011). This is sometimes regarded as a failure of transfer entropy to properly capture *causal* influences (Ay and Polani, 2008; Lizier and Prokopenko, 2010; Janzing and Balduzzi, 2012). The agent model used in this paper will exhibit this type of phenomenon, but in addition we will show that although strongly coupled components may exhibit low transfer entropy, they may still act as information conduits, hiding information transfer between more separate components.

Our model is a reactive robot designed to behave like a child swinging on a swing. As the feedback gain in the robot’s controller increases, a self sustaining oscillation is created. The agent has a simple neural model acting as its brain, which is connected to the environment via its body. The state of the agent’s neural system cannot (physically) influence the environment apart from by first affecting its body. However, we demonstrate that information transfer can take place from brain to environment *without* information transfer from brain to body. This shows how information transfer can be hidden within the agent, and revealed by its interaction with the environment.

This is the key result of this paper – information can pass through a chain of coupled systems, e.g.  $A$  to  $B$  to  $C$  such that there is a high information transfer from  $A$  to  $C$  but *not* from  $A$  to  $B$ , even though physically there is no alternative route. In the discussion at the end of the paper we will consider how similar effects occur in communication systems by way of analogy to our agent based model.

This paper is organised as follows: the next section below describes the model swinging agent and its general dynamical features. The analysis in the following section shows the information hiding phenomenon by analysing the information transfer between each component of the system. The final section discusses this result and considers the implications for the study of embodied autonomous agents in terms of information theory.

Table 1: Variables and parameters

Symbol	Type/Value	Description
$\theta$	Variable	Angle of pendulum from downward vertical
$\omega$	Variable	Angular velocity of pendulum ( $d\theta/dt$ )
$r$	Variable	Current pendulum extension
$v$	Variable	Rate of pendulum extension ( $dr/dt$ )
$u$	Variable	Force control variable – force on bob due to effector
$F_a$	Intermediate	Force on bob due to acceleration
$F_s$	Intermediate	Force on bob due to spring
$A$	Independent variable (0-80)	Motor neuron output at saturation
$g$	9.81	Acceleration due to gravity
$b$	0.3	Pendulum damping coefficient
$\rho$	2	Motor neuron sensitivity
$\phi$	20	Control parameter
$k$	100	Spring force constant
$c$	20	Spring damping ( $= 2\sqrt{k}$ for critical damping)

### Reactive swinging agent

The system studied here is a simplified model of a child swinging on a swing. The swing itself will be modelled as a rigid massless rod attached to a fixed pivot at one end with a mass (being the mass of the agent) at the other end. The agent’s motor control consists in its ability to move the mass up and down (towards and away from the pivot). There are two general ways to approach the dynamical modelling of such a system. It is possible to use a “kicked pendulum” approach where a periodic forcing function is used to perturb the mass (e.g. Belyakov et al., 2009). However it has been found that even though a pendulum can be made to swing this way, the limit cycle produced is in fact unstable, and thus this is in a practical sense impossible to achieve in the real world, suggesting that a better approach is the “self-excited” oscillator (Pinsky and Zevin, 1999; Zevin and Filonenko, 2007). Here the agent creates a positive feedback loop by adjusting the distance from the mass to the pivot point (e.g. by raising and lowering the centre of mass of the agent relative to a fixed attachment point at the end of the rod). This create a stable limit cycle as well as a resting point (where the swing is pointing straight downwards and there are no vibrations to amplify). Thus if the swing is given an initial “push”, the movement of the agent will sustain the oscillation, hence the system is described as self-excited. This approach treats the agent as a reactive system in the sense of Brooks (1986). This section provides further details on the implementation of this system.

A representation of the model is shown in figure 1. There is a massless rod with length normalised to one arbitrary unit. It makes an angle  $\theta$  with the vertical axis along which the gravitational force  $g$  applies. The “agent” consists of a mass-spring-damper system attached to the end of the rod. The mass is influenced by the gravitational force, along with

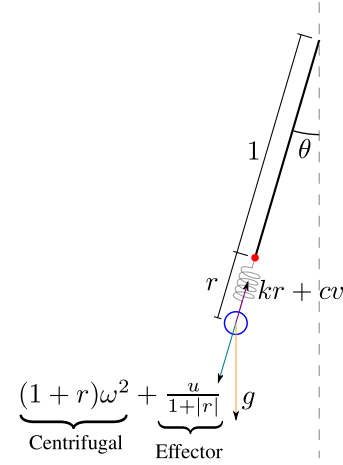


Figure 1: Spring based model of the swinging agent

the centrifugal effect of rotation and the forces created by the spring: linear contraction  $kr$  where  $k$  is a constant and  $r$  is the extension of the spring, and damping  $cv$  with  $c$  another constant and  $v = \dot{r}$  – the linear velocity of the mass in the direction of the spring. The agent creates an effector force  $u$  which acts on the mass, but this is derated according to the current absolute extension of the spring, modelling a linear motor which produces less force output when it is already extended.

The full system can be described by the following equations. Table 1 lists each of the variables and parameters used. Dots represent differentiation with respect to a non-dimensionalised time variable  $t^1$ :

<sup>1</sup>For simplicity all variables are treated as dimensionless, though the choice of  $g = 9.81$  suggests the system could be treated as a one metre long pendulum with the agent mass at one kilogram, and time in seconds.



$$\dot{\theta} = \omega \quad (1)$$

$$\dot{\omega} = -\frac{g}{1+r} \sin(\theta) - b\omega \quad (2)$$

$$\dot{r} = v \quad (3)$$

$$\dot{v} = \frac{u}{1+|r|} + F_a + F_s \quad (4)$$

$$\dot{u} = \phi(A \tanh(\rho v) - u) \quad (5)$$

The last equation describes the internal dynamics of the agent's reactive controller. The agent senses the current velocity  $v$  of its spring, and passes this through a simple sigmoidal neuron, which determines a desired output force  $A \tanh(\rho v)$  where  $A$  and  $\rho$  are parameters. The actual output force  $u$  moves towards this desired value in proportion to its current error according to the rate parameter  $\phi$ .

The acceleration of the mass in the direction of the pendulum rod  $\dot{v}$  is given by the resultant force (we assume the mass is normalised to one arbitrary unit). That is, equation 4 shows  $\dot{v}$  is the sum of the force due to acceleration  $F_a$  (i.e. gravity and centrifugal forces, equation 6) and the force due to the spring  $F_s$  (equation 7) along with the effector force described above.

$$F_a = g \cos(\theta) + (1+r)\omega^2 \quad (6)$$

$$F_s = -kr - cv \quad (7)$$

We can treat the different dynamical variables as components of either the agent or environment, and further subdivide the agent into "brain" and "body" as shown in figure 2. The intention is to treat the agent as dynamical system which is "embodied" in the sense that its overall behaviour is a result of the close coupling of the agent's body, brain and environment (Pfeifer et al., 2007a). The main sensor variable is  $v$  – the input to the neuron, though the spring extension  $r$  can also be conceptualised as a component of the agent's sensory system. The motor output is represented by  $u$ , and the environment consists of the pendulum system:  $\omega$  and  $\theta$ .

The fixed parameter values used in the following simulations are shown in table 1. The parameter  $A$  effectively controls the feedback gain and will be varied as the independent variable in what follows.

The bifurcation plot in figure 3 gives an indication of the general dynamical features of the system. These plots are obtained by recording the angular speeds at which the swing passes through the downward direction, having been initialised with a random angular velocity and the "transient" time while the system is still far from a stable cycle or point discarded. Data is obtained using Runge-Kutta integration – all results in this paper are based on an integration step size of 1/50th of a time unit, with a simulation length of 1000 time units. With  $A$  low, less than about 10, there is a

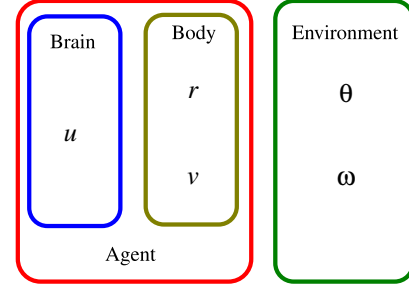


Figure 2: The agent and environment in terms of dynamical variables

single, globally stable fixed point – i.e. there is insufficient feedback for the agent to actually swing. Between feedback gains of around 10 and around 50, the agent usually swings side to side (represented in blue in the figure) – where the agent returns to  $\theta = 0$  swinging in a different direction each time. Above  $A = 30$  another stable cycle appears where the pendulum swings over the top rather than side-to-side, i.e. it returns to  $\theta = 0$  travelling in the same direction (same sign of  $\omega$ ) each time. Note that the two cycles coexist between values of  $A$  around 30 to 50, but above that only the rotating motion occurs. Finally, above  $A = 70$ , a transition to chaotic motion occurs – above this point the system will sometimes rotate and sometimes swing side to side during a single trajectory. The fixed point where the system does not swing is locally stable for values of  $A$  less than around 33, meaning that sometimes the system will tend towards resting rather than either of the limit cycles. Thus the ultimate behaviour of the system is in general dependent on the initial conditions as well as the particular value of  $A$  chosen.

We now consider a slight alteration to the model. In practice, no sensor is perfect, and thus the input to the neuron might conceivably be modelled as a stochastic variable with a slight perturbation  $\varepsilon_v$ , so equation 5 becomes:

$$\dot{u} = \phi(A \tanh(\rho(v + \varepsilon_v)) - u)$$

Assuming  $\varepsilon_v$  is small we can linearise its effect model it as a random additive perturbation on  $\dot{u}$ :

$$\dot{u} = \phi(A \tanh(\rho v) - u) + \phi A \rho \text{sech}^2(\rho v) \varepsilon_v$$

In order to practically simulate the system, it must be written as stochastic differential equations. Specifically, we convert the equation for  $u$  (which is the only variable where we directly add noise) into *Langevin equation* form:

$$du = \phi(A \tanh(\rho v) - u)dt + \phi A \rho \text{sech}^2(\rho v) \sigma dW$$

Where  $W$  represents a Wiener process, and a new parameter  $\sigma$  is introduced to control the strength of the random

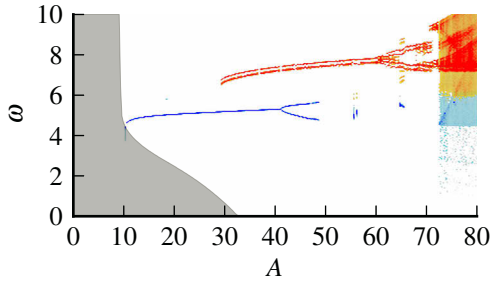


Figure 3: Bifurcation plot showing the behaviour of the system as the internal gain  $A$  is increased. Simulations are performed at each of 300 linearly spaced points between  $A = 0$  and  $A = 80$ . Plot shows absolute angular velocity of the pendulum recorded as it passes through the “downward” ( $\theta = 0$ ) plane of its state space – points in blue are returns to  $\theta = 0$  where the sign of  $\omega$  changed in between returns (i.e. the agent is swinging side to side) and points in red show returns in the same direction (the pendulum has swung over the top). The grey area shows the numerically estimated stability region for the fixed point where  $\omega = 0$  – i.e. if the system is within this region it will eventually stop swinging. Outside of this region, it will go to either a swinging or rotating stable cycle.

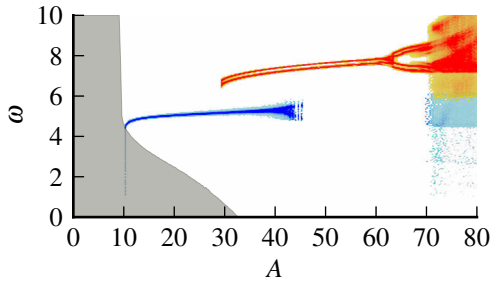


Figure 4: Bifurcations with noise  $\sigma = 0.25$

noise. This equation can be numerically solved using the stochastic strong order 1.0 Runge-Kutta algorithm. The full details of this approach including integration algorithm are found in Sauer (2012). The overall effect is that  $u$  behaves as if the neuron senses the current velocity  $v$  with additive Gaussian white noise, where the noise power is increased by increasing the newly introduced parameter  $\sigma$ . Figures 4 and 5 show the effect of increasing  $\sigma$  on the bifurcation structure – the main features remain much the same, but the crossing points are now somewhat random.

As well as making the model more “realistic”, introducing this random perturbation ensures that the system is generally ergodic, which facilitates the correct calculation of transfer entropy. Without this property, the probabilities estimated from time series data tend to make little sense (see Breiman,

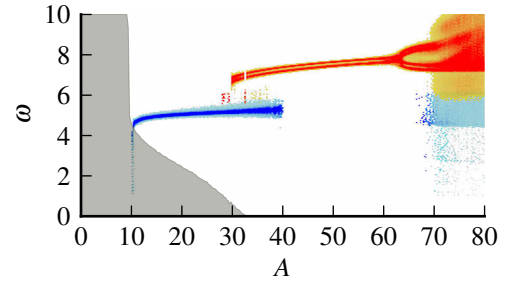


Figure 5: Bifurcations with noise  $\sigma = 0.5$ . Note that since the system is stochastic, the stability regions are not deterministically defined (close to the edge of the region shown, some trajectories may tend towards the fixed point and some towards the limit cycle depending on chance). The region shown shaded corresponds to the median stable boundary found in 20 simulation runs at each value of  $A$ .

1969, for a discussion). This slight randomness also means that even for very closely synchronised variables there will likely be at least some transfer entropy measured, as there will be a constant introduction of entropy inside the system.

### Information transfer analysis

Transfer entropy is generally defined for two time series  $X$  and  $Y$  as a relative entropy or conditional mutual information:

$$TE_{X \rightarrow Y} = \sum P(x_t, y_{t+\delta}, y_t) \log \frac{P(y_{t+\delta}|x_t, y_t)}{P(y_{t+\delta}|y_t)}$$

The data points being taken at discrete time intervals  $\delta$ , e.g.  $X = (x_{t0}, x_{t0+\delta} \dots x_{t0+n\delta})$ . The sum is taken over the support of  $P(x_t, y_{t+\delta}, y_t)$  – i.e. all possible combinations of values for the three variables. In this analysis we use the time interval  $\delta = 1$  (i.e. 50 integration steps, corresponding to approximately one quarter of a cycle).

It is problematic to calculate the transfer entropy on continuous-valued time series such as we have here. We have used symbolic transfer entropy (Staniek and Lehnertz, 2008), which uses a convenient rank transform to find an estimate of the transfer entropy on continuous data without the need for kernel density estimation.<sup>2</sup> First an embedding dimension  $m$  is chosen (we use 4), for each  $n \geq m$  we set  $\hat{x}_{t0+n\delta} = \text{rank}[(x_{t0+(n-m+1)\delta} \dots x_{t0+n\delta})]$ , where rank converts a sequence into its sort order, e.g. (0.0, 0.4, 0.3, 0.25) becomes (1, 4, 3, 2). That is, each original observation (after embedding in  $m$  dimensions) is a continuous vector ( $x_t \in \mathbb{R}^m$ ) and after transformation each observation is assigned one of the  $m!$  possible permutations

<sup>2</sup>Alternatives exist such as  $k$  nearest-neighbour methods (Kraskov et al., 2004; Evans, 2008). At this time we are not aware of a reason to prefer one method over the other in this instance.

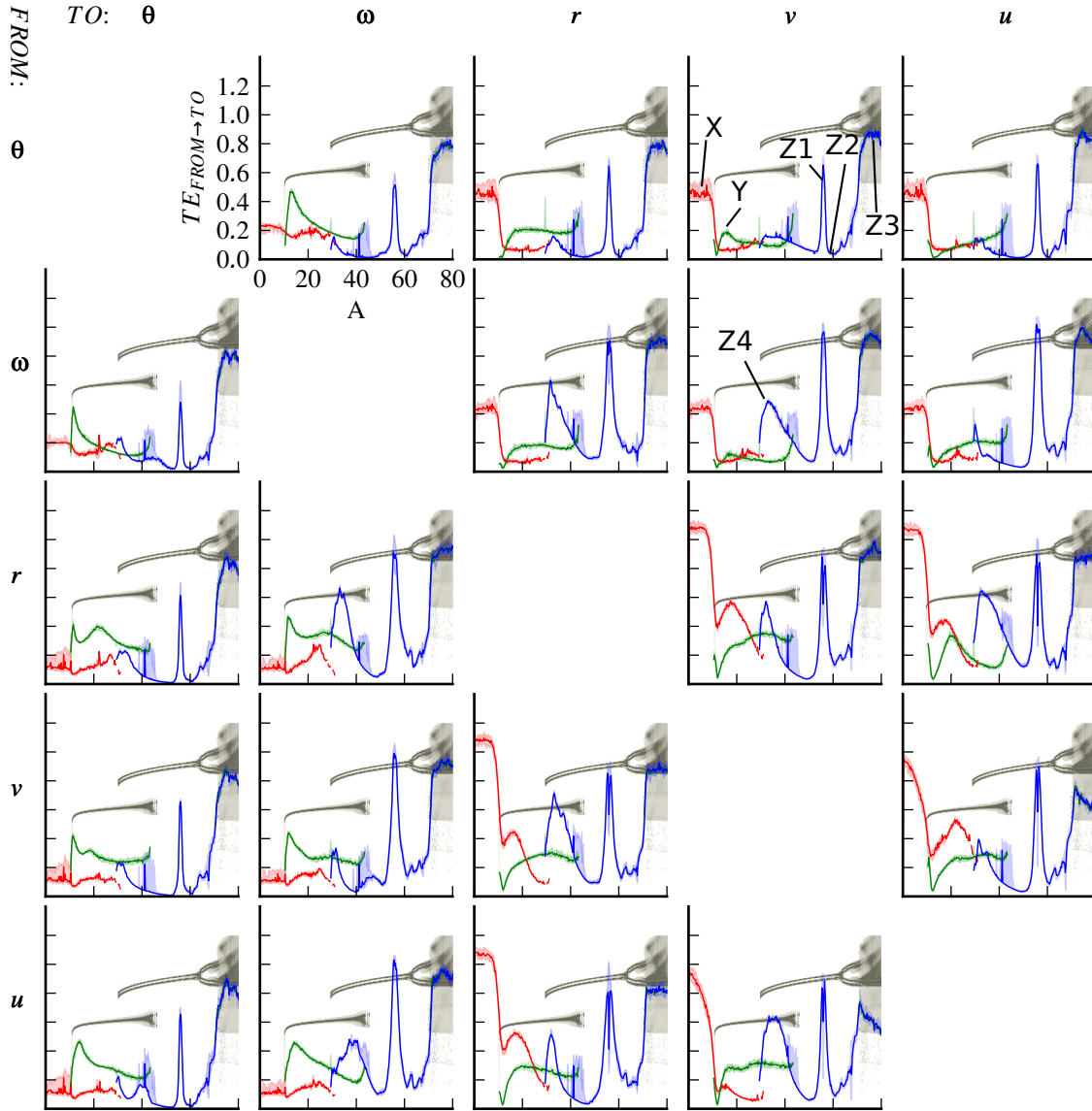


Figure 6: Symbolic transfer entropy in bits from each dynamical variable to each other one as the internal gain  $A$  is varied in the system with noise  $\sigma = 0.25$ . Note that figure 7 shows some of the same data in a form that is easier to interpret for the effects we are primarily interested in – the current figure is provided to show the context for the particular values of  $A$  chosen for re-plotting in figure 7. The background of each plot shows in grey a copy of the bifurcation diagram from Figure 4 – this is intended to help identify the correspondence between recorded transfer entropy and system behaviour. The results from 20 runs are shown after grouping by behaviour mode (color online): red for stable (non-swinging), green for side-to-side swinging and blue for rotational motion. For each behaviour the median is calculated for plotting and the shaded area around each line shows the 10th-90th percentile range where it is visible (for most values of  $A$  there was very little variation in the results). For comparison, the bifurcation plot for the system is shown in grey in the background. Some key points on the graphs are labelled in the  $\theta \rightarrow v$  and  $\omega \rightarrow v$  plots (the same features are present on some of the other plots as can be seen): at X the transfer entropy for low feedback gains (stable behaviour) is often high; at Y there is a peak in the curve for side-to-side swinging behaviour at around  $A = 12$ ; Z1 is a notable peak in the rotational swinging behaviour, which appears to correspond to some complexity in the behaviour not captured by the bifurcation diagram, Z2 is a trough at around  $A = 50$ ; Z3 and Z4 show peaks in transfer entropy which can easily be related to features of the bifurcation diagram which indicate higher complexity – the chaotic behaviour at very high gains and the behaviour close to the bifurcation point.

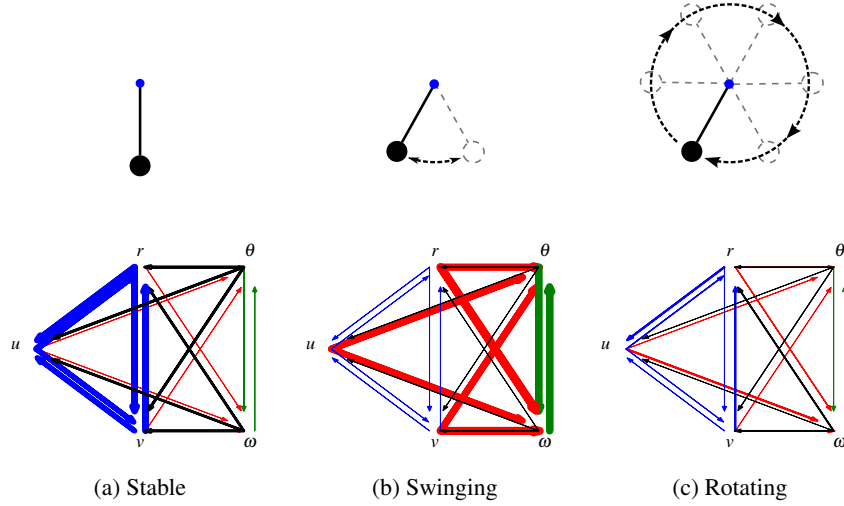


Figure 7: Median transfer entropy under three different behavioural regimes represented by arrow widths. Arrows are colored blue for information transfer within the agent (variables  $u$ ,  $r$  and  $v$ ), green for within the environment (variables  $\theta$  and  $\omega$ ), red for agent to environment and black for environment to agent. See Figure 2 for classification of variables. (a) Low feedback gain ( $A = 2.6$ ), the system cannot maintain a periodic motion and tends towards the stable state. Higher transfer entropy is seen within the agent. (b) Moderate gain ( $A = 12$ ), the agent will swing side to side. This graph illustrates the information hiding effect (see text). (c) High feedback ( $A = 50$ ), the system rotates over the top. At this value almost no transfer entropy is seen in any direction. Note that the arrow widths in (a) are 1/3rd the scale of the widths in (b) and (c) since the transfer entropy values are generally much larger in (a).

of a sequence of length  $m$ . The permutation is denoted  $\hat{x}_t$  and for ease of calculation could obviously be assigned an integer representation according to an arbitrary one-to-one mapping. The formula for symbolic transfer entropy is then

$$STE_{X \rightarrow Y} = \sum P(\hat{x}_t, \hat{y}_{t+\delta}, \hat{y}_t) \log \frac{P(\hat{y}_{t+\delta} | \hat{x}_t, \hat{y}_t)}{P(\hat{y}_{t+\delta} | \hat{y}_t)}$$

With the probabilities estimated in the natural manner for discrete variables according to frequency of occurrence, i.e.  $P(\hat{x}_t = X)$  would simply be the number of time points where  $\hat{x}_t$  is found to be  $X$  divided by the total number of observations taken.

On every experimental run, the system is initiated with all dynamical variables set to zero except for  $\omega$  which is taken uniformly at random from  $[-10, 10]$ . The first 100 time units are treated as transient non-stationary data and discarded, and the remaining 900 data points are fed to the symbolic transfer entropy calculation. This process is repeated ten times with different initial conditions, and the trajectories recorded are classified according to their final behaviour mode: resting, swinging or rotating.

The set of results in figure 6 shows all the transfer entropy values calculated for the system using a noise amplitude of  $\sigma = 0.25$ , taking each possible combination of source and target variables. This shows a few basic features of the results. We see as expected that the transfer entropy does not

straightforwardly correspond to physical coupling – there is no simple correspondence between the independent variable  $A$  and the transfer entropy value. We also see that very different patterns of information transfer are observed for the different behavioural regimes, even at the same value of  $A$ .

A simpler graphical representation of the transfer entropy is shown in figure 7. This shows the median transfer entropy for a particular behaviour at a chosen value of  $A$  as the width of an arrow pointing in the direction of information transfer. The arrows have been colour coded by the way in which they connect the brain, body and environment components.

The most striking result for our purposes is shown in figure 7b, where the feedback gain is moderate, resulting in a natural swinging behaviour. Here, the highest information transfer is along the paths coloured red which emanate from the agent (according to the classification in Figure 2) and flow towards the environment. This includes the arrows which directly connect the output of the motor neuron  $u$  to the environment variables  $\theta$  and  $\omega$ . However, there is no direct physical connection along this path since the coupling between the brain and environment is always mediated by the body. This is shown in equations 1 to 5 – the neuron output  $u$  does not appear on the right hand side of the equations for  $\dot{\theta}$  and  $\dot{\omega}$ , and hence it can only influence these variables through the intermediate coupling to its body (since the body displacement  $r$  *does* influence  $\omega$ ). Thus the informa-



tion transferred from  $u$  to  $\omega$  (shown by a thick red arrow) for example is surely carried across the chain  $u \rightarrow v \rightarrow r \rightarrow \omega$ , yet there is *low* information transfer from  $u$  to  $v$  and  $r$  (illustrated by the thin blue arrows). It is in this sense that we claim this shows a form of hidden information transfer – we know that the brain can only influence the environment by going through the body, but even when a high information transfer is measured from brain to environment, there is a smaller amount from brain to body

Figures 7a and 7c do not clearly show this phenomenon, since it is in no sense necessary for it to be present. Figure 7a seems to show the strongest connections within the agent when the feedback gain is low and the system is resting, which can be explained by the fact that the source of entropy here is the sensor noise inside the agent, and since the agent is not swinging it may move up and down, but is not likely to influence the angle of the pendulum. In figure 7 the very high feedback coupling is likely creating a highly synchronised dynamic where the observed transfer entropy is very low.

## Discussion

The key result of this work is shown in figure 7b, where during the entrained oscillatory motion of the system, the transfer entropy is shown to be higher from the brain to the environment than it is from the brain to the body, even though it is not possible for the brain to influence the environment without that influence passing through the body.

It appears that the entrained behaviour leads to a reduction in the transfer entropy measured within the agent, as can be seen by comparing the blue arrows between figures 7a and 7b. This is likely due to the close synchrony between these variables when the agent is swinging – a factor that is known to generally reduce measured transfer entropy. What is interesting is that though the swinging behaviour appears to decrease the transfer entropy within the agent, it also corresponds to increased information transfer from the agent to its environment. This is a clear demonstration of the importance of the agent’s embodiment to the information dynamics of the system – the interesting (as in measurable) interaction takes place between the agent and the environment rather than within the agent.

What we are calling *information hiding* is the way in which information coming from a variable we specifically associate with the agent’s neural system, i.e.  $u$ , appears to pass straight to the environment without having to “go through” the body, in spite of the fact that we already know that, in a physical sense, it must, since only the agent’s body is physically coupled to the environment.

It is worth attempting to gain a little intuition for how this effect is working. For an analogy that is perhaps useful in the current context, consider the simplest type of encryption system based on a symmetric key illustrated in figure 8. A key is a randomly chosen binary sequence that

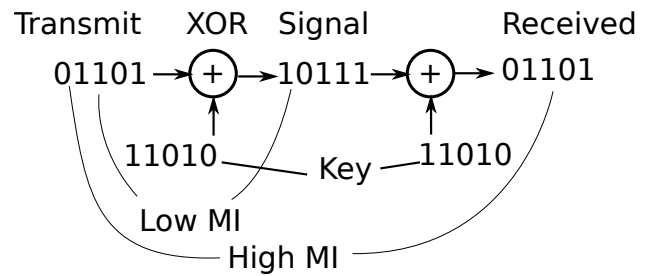


Figure 8: A simple encryption system

has been previously shared between a sending and receiving party. The sender can encrypt a message by performing the *XOR* operation bit-wise between the key and the message. However, since the key was chosen randomly, the resulting encrypted signal should be statistically independent of the transmitted message – the encryption operation appears (to anyone without the key) to flip bits of the message at random (i.e. it randomly changes some 1’s to 0’s and some 0’s to 1’s). However, with the key, it is trivial to reconstruct the original message – the same *XOR* operation is simply applied using the previously shared key. Symbolically, if we have a transmitted message  $T$ , encrypted signal  $S$  and received signal  $R$  then we have a very low  $I(T; S)$  yet high  $I(T; R)$ . Though expressed in terms of mutual information rather than transfer entropy, this is essentially the same information hiding phenomenon as we have been discussing. Indeed if we assume the individual bits of the message and the key are independent of each other then  $TE_{T \rightarrow S} = I(T; S | \text{history}(S)) = I(T; S)$  and so on.

The information hiding process can thus be seen as a message being obscured by at some point and later reconstructed. In the example above this function is performed by the encryption system and is dependent on having a piece of secondary data (the key) shared between the two endpoints via some alternative channel to the main signal path. Of course, the encryption system is carefully designed to achieve this – it requires the deliberate sharing of the key. However, comparable processes have been found relying only on chaotic synchronisation: Cuomo and Oppenheim (1993) demonstrated that synchrony between a pair of coupled Lorenz attractor systems can be used to “hide” information in a similar way.<sup>3</sup> Their experiment suggests that it is plausible that information could be hidden by a dynamical process such as the one studied here without the need for the deliberate design of an encryption system.

This phenomenon should not be viewed as information being completely lost to the world and then coming back

<sup>3</sup>Note that this system is not generally regarded as computationally secure as an encryption mechanism since the reconstruction circuit (which effectively serves as the “key”) can be relatively easily inferred using attractor reconstruction on the transmitted signal.

– rather it is simply hidden and then reconstructed by the action of some dynamical system. We have interpreted information here as a statistical summary of collected data – not as a physical quantity that exists in the world.

We have said little explicitly about causation, though of course to say that the brain must influence the environment via the body suggests a causal interpretation. Recent work has studied the relationship between transfer entropy and causal inference in part motivated by phenomena similar to the one described here (e.g. Ay and Polani, 2008; Lizier and Prokopenko, 2010). Information theory has also been applied successfully in the context of embodied systems (e.g. Ay et al., 2008; Klyubin et al., 2008). Both of these connections are relevant: can information hiding as presented here be useful in any sense as a guide to causal inference? How should the current case study be connected to wider theories of embodied behaviour? We aim to address these questions in future work.

### Acknowledgements

Thanks to Andy Philippides and Lucas Wilkins for helpful discussions around the content of this work.

### References

- Ay, N., Bertschinger, N., Der, R., Güttler, F., and Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *The European Physical Journal B*, 63(3):329–339.
- Ay, N. and Polani, D. (2008). Information flows in causal networks. *Advances in Complex Systems*, 11(01):17.
- Barnett, L. (2009). Granger Causality and Transfer Entropy Are Equivalent for Gaussian Variables. *Physical Review Letters*, 103(23):1–10.
- Belyakov, A. O., Seyranian, A. P., and Luongo, A. (2009). Dynamics of the pendulum with periodically varying length. *Physica D: Nonlinear Phenomena*, 238(16):1589–1597.
- Bertschinger, N., Olbrich, E., Ay, N., and Jost, J. (2008). Autonomy: an information theoretic perspective. *Bio Systems*, 91(2):331–45.
- Breiman, L. (1969). *Probability and Stochastic Processes: With a View Toward Applications*. Houghton Mifflin Company, Boston, MA.
- Brooks, R. (1986). A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, (1):14–23.
- Cuomo, K. and Oppenheim, A. (1993). Circuit implementation of synchronized chaos with applications to communications. *Physical Review Letters*, 71(1):65–68.
- Evans, D. (2008). A computationally efficient estimator for mutual information. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 464(2093):1203–1215.
- Janzing, D. and Balduzzi, D. (2012). Quantifying causal influences. *arXiv preprint arXiv:1203.6502*, pages 1–23.
- Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2008). Keep your options open: an information-based driving principle for sensorimotor systems. *PLoS ONE*, 3(12):e4018.
- Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Physical Review E*, 69(6):16.
- Lizier, J. T. and Prokopenko, M. (2010). Differentiating information transfer and causal effect. *The European Physical Journal B*, 73(4):605–615.
- Lungarella, M. and Sporns, O. (2006). Mapping information flow in sensorimotor networks. *PLoS computational biology*, 2(10):e144.
- Moioli, R. C., Vargas, P. A., and Husbands, P. (2012). Synchronisation effects on the behavioural performance and information dynamics of a simulated minimally cognitive robotic agent. *Biological cybernetics*, pages 407–427.
- Pfeifer, R., Lungarella, M., and Iida, F. (2007a). Self-organization, embodiment, and biologically inspired robotics. *Science (New York, N.Y.)*, 318(5853):1088–93.
- Pfeifer, R., Lungarella, M., Sporns, O., and Kuniyoshi, Y. (2007b). On the information theoretic implications of embodiment - principles and methods. In Lungarella, M., Iida, F., Bongard, J., and Pfeifer, R., editors, *50 Years of Artificial Intelligence*, volume 4850 of *Lecture Notes in Computer Science*, pages 76–86, Berlin / Heidelberg. Springer.
- Pinsky, M. and Zevin, A. (1999). Oscillations of a pendulum with a periodically varying length and a model of swing. *International Journal of Non-Linear Mechanics*, 34(1):105–109.
- Pitti, A., Lungarella, M., and Kuniyoshi, Y. (2009). Generating spatiotemporal joint torque patterns from dynamical synchronization of distributed pattern generators. *Frontiers in Neuro-robotics*, 3(2).
- Sauer, T. (2012). Numerical solution of stochastic differential equations in finance. In *Handbook of Computational Finance*.
- Schmidt, N. M., Hoffmann, M., Nakajima, K., and Pfeifer, R. (2012). Bootstrapping Perception Using Information Theory: Case Studies in a Quadruped Robot Running on Different Grounds. *Advances in Complex Systems*, 16:1250078.
- Schreiber, T. (2000). Measuring information transfer. *Physical Review Letters*, 85(2):461–464.
- Staniek, M. and Lehnertz, K. (2008). Symbolic Transfer Entropy. *Physical Review Letters*, 100(15):1–4.
- Thorniley, J. (2011). An improved transfer entropy method for establishing causal effects in synchronizing oscillators. In Lenaerts, T., Giacobini, M., Bersini, H., Bourguin, P., Dorigo, M., and Doursat, R., editors, *ECAL 2011: Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*. MIT Press.
- Williams, P. and Beer, R. D. (2010). Information Dynamics of Evolved Agents. In *From Animals to Animats 11*, pages 38–49. Springer, Berlin / Heidelberg.
- Zevin, A. and Filonenko, L. (2007). A qualitative investigation of the oscillations of a pendulum with a periodically varying length and a mathematical model of a swing. *Journal of Applied Mathematics and Mechanics*, 71(6):892–904.

# Impact of Personal Fabrication Technology on Social Structure and Wealth Distribution: An Agent-Based Simulation Study

Amber Ferger<sup>1</sup>, Wai Fai Lau<sup>1</sup>, Philipp Ross<sup>1</sup>, Wyman Zhao<sup>1</sup>, Hiroki Sayama<sup>1\*</sup>, Steen Rasmussen<sup>2,3</sup>

<sup>1</sup>Binghamton University, State University of New York, Binghamton, NY, USA

<sup>2</sup>Center for Fundamental Living Technology, University of Southern Denmark, Odense, Denmark

<sup>3</sup>Santa Fe Institute, NM, USA

\*Corresponding author: sayama@binghamton.edu

The recent surge of the “Maker” movement (Anderson 2012) is largely driven by the increasing availability of the personal fabrication technology such as 3D printing (Lipson & Kurman 2013). Widespread diffusion and adoption of personal fabricators based on von Neumann style general purpose constructors are considered as one of the central possibilities of living technology (Bedau et al., 2010; Rasmussen et al. 2011). This is expected to cause a major shift of design and manufacturing power from large firms to individuals, in ways similar to how personal computers and information technology changed information production and dissemination in our society over the last few decades.

Personal fabricators occupy an interesting position in the network of goods and products from an artificial chemistry perspective (Dittrich, Ziegler & Banzhaf 2001). The whole manufacturing system in human civilization can be understood as a huge metabolic network in which reactants and catalysts are goods (raw materials and products) and fabricators (production tools ranging from human hands, hammers and knives to advanced computers and large-scale factories), respectively (Becker et al. 2013). In this context, the role of personal fabricators is close to that of general-purpose catalysts, such as ribosomes in biological systems. Such general-purpose catalysts tend to have high complexity and slow reaction rates, yet they can produce a great variety of complex products.

The emergence of ribosomes and gene-protein translation mechanisms in the history of life enhanced diversity, functionality and complexity of biomolecular machines significantly, clearly marking one of the major evolutionary transitions (Maynard-Smith & Szathmari 1997). This observation naturally leads one to ask the following question:

*What kind of societal transition may occur due to the rise of personal fabricators?*

This is not a trivial question to answer because of the differences between biological and socio-economical systems. Unlike biological cells, modern socio-economical systems are largely driven by individuals who have conflicting personal and financial interests and strategically determined behaviors imbedded in a market driven economical system. Such behavioral complexities of anthropomimetic agents are often omitted in computational social simulations, though they would be necessary in order to capture relevant socio-economical dynamics.

To investigate the potential societal impact of the rapidly emerging personal fabrication technology, we developed an agent-based simulation of designing-manufacturing-economy dynamics. In this model, agents design and produce goods using other goods as materials based on their knowledge of manufacturing processes, and then trade the products with other agents, in order to maximize their own utility (largely determined by monetary profit). The system consists of two main components: (i) *the static universal product network* made of all  $n$  possible goods using  $m$  materials in different ways determined by their connections and (ii) *the dynamic economy* made of agents and their realized markets, which are where goods are actually traded.

*The static universal product network* represents the global set of all manufacturing processes possible (including not yet realized) in the simulated world. It is randomly generated using a heuristically designed algorithm as a bipartite network made of two types of nodes: reactants (raw materials and products) and reactions (production processes). Each reaction combines multiple reactants and produces another reactant at a certain rate. In this network, fabricators can be identified as catalytic reactants that do not increase or decrease in number through a reaction process.

Each catalytic reactant (fabricator) has its inherent complexity and utility values assigned to it. The complexity of a product is always bounded by the complexity of the fabricators and products used in its production process. Fabricators with high complexity, slow reaction rates but great universality (i.e., ability to catalyze a great number of reactions) represent general purpose personal fabricators. In contrast, fabricators with medium complexity, fast reaction rates and high specificity represent large-scale mass-production factories. Fabricators with low complexity, slow reaction rates and high universality represent primitive manufacturing tools such as hand tools. Agents are initially equipped with the fabricator with lowest complexity to start with.

*The dynamic economy* consists of agents and their markets. Each agent has the following properties: (1) amount of money it has, (2) inventory and price of goods they own (including fabricators), (3) utility associated with each product in the product network, and (4) the combined utility of all of the products in their inventory. At each time step, an agent randomly generates a finite number of possible actions (which goods to produce, which goods to purchase, etc.) and assesses

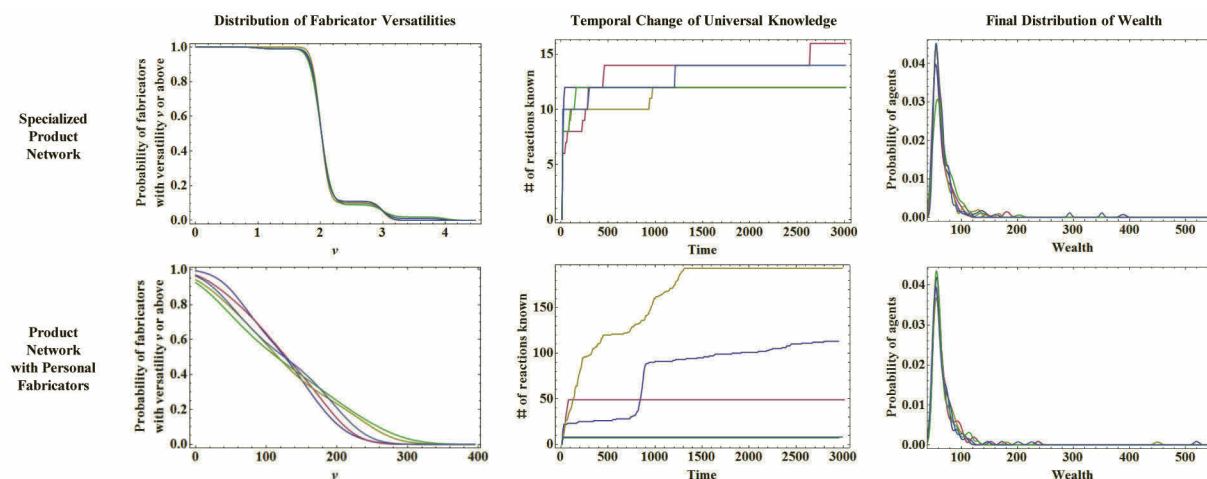


Figure 1: Simulation results. Results of five independent runs are shown in each plot. Left column: Complementary cumulative distributions of fabricators' versatilities. Top (across all columns): Results with a specialized product network without versatile personal fabricators. Bottom (across all columns): Results with a production network with versatile personal fabricators.

them to choose one that will increase its overall utility most as the next action.

Agents have partial knowledge about the universal product network (i.e., subgraph of the network), which is shared amongst all agents. At any given time step, there is a small probability by which an agent discovers (and implements) a new reaction in the universal product network and add this information to the communal knowledge base. This simulates gradual accumulation of innovations in society over time. An agent is only able to produce products that are included in the communal knowledge base.

Using the agent model described above, we conduct simulations with the abundance of personal fabricators in the universal product network varied as the experimental input. More specifically, we test two different scenarios of the universal product network: (1) a specialized product network where the fabricators' versatilities (i.e., number of products a fabricator can produce) are quite low and homogeneous (Fig.1, top left), and (2) a product network where the fabricators' versatilities show a fat-tail distribution in which some fabricators can produce a great number of different products (representing the possibility of personal fabricators; Fig. 1, bottom left).

Our preliminary simulation results show an interesting difference in technological development between the two scenarios, as shown in Fig. 1 (middle and right). With the specialized product network, after 3,000 time steps, agents tend to discover only a small number of reactions (top middle). In contrast, with the product network with versatile personal fabricators, agents can discover an order of magnitude larger number of reactions (bottom middle), and they often achieve innovative breakthroughs (seen as rapid increases of curves). The final distributions of wealth are found to be similar in both conditions, however (bottom right).

We note that our simulation is still preliminary and limited in several aspects. First, to include the key components the complexity of the current simulation is quite high as it implements a number of assumptions involved in designing-manufacturing-economy dynamics. This makes it rather difficult to explore, calibrate and validate experimental settings.

Second, the economic rules currently used to determine the prices of goods are simplistic and could be improved by implementing more well-established economics theories. Third, the incentives for agents to discover and produce new innovative products are currently given by inherent utility assigned to each potential product, which may not be a valid assumption to make. We are currently working on a simplification, revision and more thorough validation of our simulation.

**Acknowledgements:** This work was partly conducted as a senior design project of the Department of Bioengineering at Binghamton University. We thank Dr. George Catalano, Professor of Bioengineering and instructor of the senior design course sequence, for his generous support and guidance over the past year. We also thank Dr. Andreas Pape for his advice on the modeling of agents' economic decision making.

## References

- Anderson, C. (2012). *Makers: The New Industrial Revolution*. Random House Business Books.
- Lipson, H., & Kurman, M. (2013). *Fabricated: The New World of 3D Printing*. Wiley.
- Bedau M.; Hansen, P.G.; Parke, E. & Rasmussen, S. (2010) *Living Technology: 5 Questions*, Automatic Press / VIP. p 236
- Rasmussen, S., Albertsen, A., Fellermann, H., Pedersen, P. L., Svaneborg, C., & Ziöck, H. (2011). Assembling living materials and engineering life-like technologies. In *Proceedings of GECCO 2011* (pp. 15-20). ACM.
- Dittrich, P., Ziegler, J., & Banzhaf, W. (2001). Artificial chemistries-a review. *Artificial life*, 7(3), 225-275.
- Becker, T., Meyer, M., Beber, M. E., Windt, K., & Hütt, M. T. (2013). A comparison of network characteristics in metabolic and manufacturing systems. In *Dynamics in Logistics* (pp. 141-150). Springer Berlin Heidelberg.
- Maynard-Smith, J., & Szathmari, E. (1997). *The Major Transitions in Evolution*. Oxford University Press.



# Adapting the Bak-Sneppen Model to a Dynamic and Partially Connected Grid of Hierarchical Species

Carlos M. Fernandes<sup>1,3</sup>, Juan L.J. Laredo<sup>2</sup>, J.J. Merelo<sup>3</sup>, Carlos Cotta<sup>4</sup>, Agostinho C. Rosa<sup>1</sup>

<sup>1</sup>Institute for Systems and Robotics, Technical University of Lisbon, Portugal

<sup>2</sup>University of Luxembourg, Luxembourg

<sup>3</sup>Dept. Arquitectura y Tecnología de Computadores, ETSII Informática, Granada, Spain

<sup>4</sup>Dept. Lenguajes y Ciencias de la Computación, ETSI Informática, Málaga, Spain  
cfernandes@laseeb.org

## Abstract

This paper describes and investigates a swarm intelligence system with similarity-oriented behavioral rules, hierarchical clustering and evolution by random mutation. The evolutionary scheme is based on the Bak-Sneppen model of co-evolution between interacting species. The swarm of species, in this case, is randomly distributed on a 2-dimensional grid of nodes. The number of nodes is larger than the swarm size and the species are allowed to move on the grid. The rule that defines the movement of the species through the grid is based on the similarity between the species' fitness values and the ranking of those same values within the entire population. Meanwhile, the fitness values are modified using the rules of a 2-dimensional Bak-Sneppen model. The system is intended to be a framework for metaheuristics with spatially structured populations and we show that it displays the desired characteristics for that purpose. Furthermore, these characteristics emerge as global patterns from the local interaction of the species. Without requiring the tuning of control parameters to precise values, the system seems to self-organize into a critical state between randomness and order.

## Introduction

*Self-organization* is a concept that includes a wide range of systems and dynamics. It is used in the realm of physics, chemistry, mathematics, biology and even in social sciences. In general, the term refers to a process through which a system increases its complexity without any external action. Although the complexity sciences have not yet devised a mathematical language that explains the origins and dynamics of self-organization, it may be stated that self-organization describes the property of systems whereby unexpected global patterns emerge from local rules. This paper presents a self-organized model of a population of simple entities that displays coherent global behavior emerging from local rules. The model was designed with the main objective of being applied as a dynamic and self-regulated base-structure for non-panmictic (or structured structured) population-based metaheuristics. The resulting system is a type of swarm intelligence – see Kennedy and Eberhart (2001).

Swarm intelligence algorithms are self-organized systems in which unsophisticated distributed entities interact locally, causing global patterns to emerge. The interaction may be

restricted to the communication between the entities, or it may use an environment as a medium for that communication. When the entities interact with (and via) the environment, the system is said to be *stigmergic*, a term introduced by Grassé (1959) to describe the ability of social insects in using the environment as a communication medium.

Fernandes et al. (2012) have recently described a new swarm intelligence discrete system with stigmergic local rules. The system consists of a population of  $n$  simple individuals (or particles) moving and interacting on a 2-dimensional grid of nodes. Stigmergy is modeled by providing the particles with the capacity of depositing and following *marks* that carry information about the particle. The structure is defined by local spatial neighborhood and results in a partially connected and dynamic grid of individuals. Each individual is assigned with a random value in the range  $[0,1]$ . This value is called *fitness*.

The motivation behind the work by Fernandes et al. (2012) is to create a dynamic framework for non-panmictic Evolutionary Algorithms (EAs), as defined by Tomassini (2005). EAs belong to a class of metaheuristics based on the Darwinian theories of evolution by natural evolution that use a population of possible solutions (individuals) to a problem. The population evolves by selection, recombination and mutation towards optimal regions of the search landscape. In panmictic EAs, every individual is allowed to interact with every other individual in the population. However, large-scale problems or deceptive functions with multiple local optima may require other type of structures. Therefore, in recent years, non-panmictic EAs, also known as *spatially structured EAs* (see Tomassini (2005)), are gaining increasing attention by the community. This class of EAs restricts the interaction according to a pre-defined or evolving structure that connects the population of solutions. They permit to control the genetic diversity of the population and avoid premature convergence, but they also require extra designing and tuning efforts. In addition, the chosen structure affects the connectivity and the performance of the algorithm.

One possible approach to overcome the rigid connectivity of the traditional structures without being trapped in complicated network design is to use the self-organizing and emergent properties of complex adaptive systems. The work by Fernandes et al. (2012) is an attempt to model the desired

characteristics of a dynamic and self-regulated population structure for non-panmictic EAs. In fact, complex properties, such as dynamic clusters of particles displaying pink noise patterns, have been observed while testing the model. However, the experiments in Fernandes et al. (2012) are restricted to a stationary version of the model, i.e., the fitness values of the individuals do not change during the run.

This paper extends the study by Fernandes et al. (2012) and investigates the behavior of the system when populations of time-varying fitness values interact on the grid and generate the structure. The rules for varying the fitness values were taken from the Bak-Sneppen model of co-evolution between interacting species, a complex system proposed by Bak and Sneppen (1993): in each time-step, the fitness value of the worst individual and the fitness values of its neighbors (if any) are replaced by random values in the range  $[0,1]$ . In other words, the worst individual and its adjacent neighbors in the habitat are mutated.

The Bak-Sneppen model is an example of Self-Organized Criticality (SOC), a theory that has been proposed by Bak et al. (1987) for explaining a class of systems that self-organize into a critical state without requiring the tuning of control parameters. When in the critical behavioral region, these systems display typical signatures, such as scale-invariance, power-law relationships between events and their intensity (or duration) and output variables with pink noise power spectrum.

The Bak-Sneppen model has all the above referred signatures. Like other SOC systems, it doesn't require parameters that need to be tuned. Furthermore, its global behavior can be described as a population of fitness values that evolve during the run. The average fitness of the population tends to grow and the gap  $G(t)$  of the system, which is the maximum of the minimum fitness before time-step  $t$ , is increased during the run until it reaches a specific range (that depends on the topology of the population). These characteristics make the Bak-Sneppen a good candidate for being implemented on the framework proposed by Fernandes et al. (2012) in order to investigate if the behavior observed in the stationary version is maintained in a population of time-varying fitness values. Moreover, the resulting model provides the opportunity to study a version of the Bak-Sneppen model that, to the extent of our knowledge, has not yet been proposed. This new version is characterized by a dynamic topology and by the self-regulated and hierarchical clustering of species.

In this paper, the experiments were designed for describing the properties of the new system, for analyzing the system's behavior in search for complexity and self-organization signatures, and for testing the robustness of the system to changes in the fitness distribution of the population.

The remainder of the paper is structured as follows. The following section addresses SOC and describes the original Bak-Sneppen model. Then, the proposed system is described and contextualized within the current research on spatially structured populations. The subsequent section describes the experiments and the system's dynamic behavior. The final section concludes the paper and outlines future lines of research.

## SOC and The Bak-Sneppen Model

SOC is a critical state formed by self-organization in a long transient period at the border of order and chaos. While *order* means that the system is working in a predictable regime where small disturbances have only local impact, *chaos* is an unpredictable state very sensitive to initial conditions or small disturbances. In complex adaptive systems, complexity and self-organization usually arise at that transition region between order and chaos, or *on the edge of chaos*, as it is sometimes stated. SOC systems are dynamical with a critical point at the region between order and chaos as an attractor. However, and unlike many physical systems, which have a parameter that needs to be tuned in order to obtain the critical state, SOC systems are able to self-tune to the critical point.

In a SOC system, small disturbances can lead to the so-called *avalanches*, that is, events that are spatially or temporally spread through the system. This happens independently of the initial state. Moreover, the same perturbation may lead to small or large avalanches, which in the end will display a power-law proportion between the size of the events and its abundance.

This means that large (catastrophic) events may hit the system from time to time and reconfigure it. These power-law relationships between the size of the events and their frequency are widespread in Nature. Earthquake distribution, for instance, follows the Gutenberg-Richter law, which is a power-law proportion between the magnitude of the earthquakes that occurred in a specific area during a specific period of time, and the frequency of those earthquakes. *Pink noise*, or  $1/f$  noise, also displays power-law behaviour (as opposed to *white noise*, which is chaotic).

One may distinguish three types of power-laws arising from physical systems. For instance, the power spectral density distribution (like the pink noise) is described by:

$$P(f) \propto \frac{1}{f^\alpha} \quad (1)$$

where  $f$  is the frequency,  $P(f)$  is the power of that frequency and  $\alpha$  is a real number between 0 and 2.0, but usually close to 1.0. If  $\alpha = 0$  then  $P(f)$  is named white noise; if  $\alpha = 2.0$  then it is named red noise or Brownian noise; when  $\alpha = 1.0$  then the function  $P(f)$  describes pink noise. In general, this function describes which frequency is the most dominant in the temporal behaviour of the system under consideration: the power spectral density is just the square of the Fourier transform of the signal under consideration.

Another power-law arises in size distributions (like the Gutenberg-Richter law, for instance):

$$N(s) \propto \frac{1}{s^\tau} \quad (2)$$

where  $s$  is the size of an event (or magnitude) and  $N(s)$  reflects a distribution of frequency of such events.

A third kind of power-law is identified in the temporal distribution of events, where  $\tau$  is either the duration of the event, or the time between events, as described by equation (3):

$$N(\tau) \propto \frac{1}{\tau^\nu} \quad (3)$$

SOC may be the common link between a wide range of natural phenomena operating at the region between order and chaos that exhibit these power-law relationships, a scale-

invariant behavior that does not need to be tuned. The first system where SOC was identified is a cellular automaton called *sand pile* and it is described by Bak et al. (1987). Later, Bak and Sneppen (1993) introduced the model of co-evolution between interacting species: the Bak-Sneppen model.

In nature, different species in the same eco-system are related through several features (food chains, for instance). They co-evolve, and the extinction of one species affects the species that are related to them, in a chain reaction that can reach huge proportions. Fossil records suggest that the size of extinctions events is in power-law proportion to its frequency. It is also known that the biological history of life on Earth is punctuated by catastrophic extinction events. The Bak-Sneppen model aims at understanding and explaining the mechanisms underlying mass extinction. It consists of a number of species, each one with a fitness value assigned and each one connected to other species (neighbors). Every time step, the species with the worst fitness and its neighbors are eliminated from the system and replaced by individuals with random fitness.

This description may be translated to a mathematical model. The system is defined by  $n^d$  fitness numbers  $f_i$  arranged on a  $d$ -dimensional lattice (ecosystem) with  $n$  cells. At each time step, the smallest  $f$  value and its  $2 \times d$  neighbours are replaced by uncorrelated random values drawn from a uniform distribution (in other words, the worst species is removed from the population and its neighbors are mutated). The system is thus driven to a critical state where most species have reached a fitness above a certain threshold and the avalanches produce non-equilibrium fluctuations in the configuration of the fitness values. The complex behavior is observed even in the one-dimensional case, where species are arranged in a chain, and each one has two neighbors.

Since its proposal, the model has been thoroughly investigated by the community and several extensions and modifications have been described. In the seminal paper by Bak and Sneppen, the research is focused on the 1-dimensional version of the system. Higher dimensional models have been since then investigated. De los Rios et al. (1998), for instance, study the high dimensional Bak-Sneppen model ( $d \geq 2$ ) and conclude that the system shows a rich behavior with four qualitatively different regimes as a function of dimensionality:  $d \leq 2$ ,  $2 < d < 4$ ,  $4 \leq d < 8$  and  $d \geq 8$ .

In this paper, we have used the rules of a Bak-Sneppen model with  $d = 2$ . However, the resulting system is not a standard 2-dimensional Bak-Sneppen model. In our model, the position of species is dynamic and the grid is partially connected, i.e., each species may have four or less species in its von Neumann neighborhood. This leads necessarily to a different behavior and the dynamics observed in the 2-dimensional model may not occur. However, we are mainly interested in the behavior of the proposed system as a potential framework for spatially structured EAs and therefore we search for signatures of dynamic clustering and robustness to changes. A theoretical analysis and empirical validation of the Bak-Sneppen model for determining critical exponents and the gap function is left for future work.

## The System

The proposed framework is a discrete system with a swarm of heterogeneous individuals controlled by a set of local rules. The rules define the actions of a population of  $n$  particles that move on a 2-dimensional toroidal grid of nodes with size  $X \times Y$ . In each time-step, every particle tries to move to a neighboring node. The rules that model the system are the following.

At  $t = 0$ , the particles are assigned a random *fitness* value in the range  $[0,1]$  and then randomly distributed in a  $X \times Y$  grid of nodes. Then, at each time-step, each particle moves to an adjacent free node (if any), leaving a mark with information on its status in the previous node. In this paper, the status is the fitness of the particle. The particles decide where to go by inspecting their Moore neighborhood. If there are no free nodes in the neighborhood (i.e., all the cells are occupied by particles), the particle stays in that same node until the next iteration. If there are free cells, the particle checks for marks. If it finds no marks, it just randomly chooses a destination node between the free neighboring nodes. If marks are found with better fitness than the particle's fitness, the particle moves to the node with the mark that minimizes the difference between its fitness and the fitness on the mark. Whenever a particle changes its position, it leaves a mark in its previous location. The marks only remain in the *habitat* for a time-step. In summary, communication by dropping and following information is the base-rule of the proposed system. The system is modeled with a stigmergic behavior.

The particles are ranked according to their fitness. This strategy is imposed with the objective of establishing a hierarchy in the self-organization of the clusters: worst particles tend to follow better particles (the better individuals are *leading the way*).

In each time-step (which comprises the update of every particle's position), the particle with lowest fitness is mutated (i.e., its fitness is replaced by a random value with uniform distribution within the range  $[0,1]$ ), as well as the fitness of its neighbors. The position of the neighbors is defined by the von Neumann neighborhood of the particle (with range 1). This is the standard Bak-Sneppen model on 2-dimensional habitats. The only difference is that in this case the number of neighbors of the worst particle that are also mutated is not necessarily  $2 \times d = 4$ . This is the maximum number of individuals that are mutated. If the worst particle is isolated (if there are no particles in its von Neumann neighborhood) there are no more mutations in that time-step except for the particle itself. Therefore, in each time-step,  $p$  particles are mutated, with  $p \leq 1 + 2 \times d$ .

This basic set of rules drives the system towards a dynamic global pattern that displays signs of self-organization. A structure of particles, formed by clusters and paths, emerges on the habitat. However, these clusters are far from being static and, in a few generations, the distribution of the whole swarm may change dramatically (while maintaining a typical configuration of clusters and paths). The swarm's behavior is not ordered (nor chaotic). Please remember that convergence to a behavioral region between order and chaos is a signature of self-organization.

---

**Algorithm**


---

1. Randomly place  $n$  particles in a grid of node with size  $X \times Y$
  2. Randomly attribute a fitness value to each particle
  3. Find the particle with the lower fitness value. Mutate its fitness and the fitness of its neighbors (von Neumann neighborhood).
  4. Rank the particles by increasing fitness
  5. For each particle do
    6. check Moore neighborhood for marks and other particles
    7. if no marks in the neighborhood
      8. move to a free cell in the neighborhood (if any)
    9. if there are marks in the neighborhood
      10. move to the site of the nearest fitness mark which is better than its own fitness
      11. leave a mark in the previous site
      12. erase the mark in the new site
  13. if stop criteria not met return to 3
- 

Besides dynamic clusters, there are other signatures that suggest that the model comprises a *hidden order* that emerges from local rules. The following section tries to detect and describe those signatures under static and dynamic populations.

Please note that the only parameters that need to be set are the population size  $n$  and the grid size. If the ratio between  $n$  and the grid size is set within a specific range (large enough to allow communication between the particles, while not so large so that the particles hardly move on the grid), the system self-organizes without requiring the tuning of control parameters. The following section shows that dynamic global patterns emerge within a wide range of population and grid size values. But first, let us discuss the motivation behind the proposed model.

### Motivation

Genotypic representation, operators, selection schemes and population size are typical panmictic EAs moduli that require design choices. However, a population structure may be also introduced in the design scheme of this class of algorithms. This structure specifies a network of acquaintances for individuals to interact, that is, mating or selection is restricted to neighborhoods within the network structure. Spatially structured EAs include fine-grained approaches such as cellular EAs and course-grained approaches such as island models.

The initial objective of spatially structured EAs was to develop a framework for studying massive parallelization – see Gordon and Whitley (1993). Afterwards, the need to provide traditional EAs with a proper balance between exploration and exploitation motivated several lines of research that explore the potentiality of different population structures in maintaining genetic diversity. Population structures were primarily devised as static regular lattices: every individual has a fixed number of potential interaction partners. Later on, complex population structures have been also studied – by Giacobini et al. (2005) and Payne and Eppstein (2006), for instance. However, these standard *cellular EAs* have some drawbacks: synchronicity (in most cases) and a strong dependence on the problem since the

genetic diversity promoted by a prefixed topology is uncorrelated to the problem structure.

Dynamic population structures have only recently raised the interest of researchers. To the extent of our knowledge, only few works address explicitly the issue of dynamic population structures in cellular EAs. Alba and Dorronsoro (2005) dynamically change the ratio that defines the neighborhood of interaction. Since the ratio may affect selection pressure, the authors analyze its influence on the balance between exploration and exploitation. However, the base-structure of the cellular EA (i.e. a grid lattice) is maintained throughout the run.

Whitacre et al. (2008) focus on two important conditions missing in EA populations: a self-organized definition of locality and interaction epistasis. With that purpose in mind, they propose a dynamic structure and conclude that these two features, when combined, provide behaviors not observed in the canonical EAs or traditional spatially structured EAs. The most noticeable change in the behavior is an unprecedented capacity for sustainable coexistence of genetically distinct individuals within a single population. The authors state that the capacity for sustained genetic diversity is not imposed on the population; instead, it emerges as a natural consequence of the dynamics of the system.

Laredo et al. (2010) propose a framework for EAs based on peer-to-peer networks (see Steinmetz and Wehrle (2005) for a survey on peer-to-peer networks). Within a simulated environment, they model the dynamics of real networks and conclude that their system is able to achieve better performance than traditional EAs on a wide range of problems, while being scalable and resilient to the volatility of nodes in the network.

The work by Fernandes et al. (2012), extended in this paper with a Bak-Sneppen model, has some minor similarities to that by Whitacre et al. (2008), since the structural characteristics of complex systems within an EA population are also recreated. However, while in Whitacre et al. (2008) the structure co-evolves with the EA until it reaches a stable self-organized state, the system described here does not converge to rigid or nearly-rigid state. Instead, it aims at a system working in a critical state where links are frequently created and destroyed and where new emergent patterns appear at high rate.

We demonstrate that the proposed system has indeed emergent properties that could prove useful for spatially structured EAs, or other spatially structured population-based metaheuristics. In this paper, the dynamics of the system and its self-organizing behavior are studied under dynamic populations: the fitness values vary through the run according to the rules of the Bak-Sneppen model. Such dynamics are intended to model the behavior of EAs on the proposed framework. Therefore, it is expected that the outcome of the experiments can provide information on the self-organizing properties of the system and on the limits of those properties.

## Experiments and Discussion

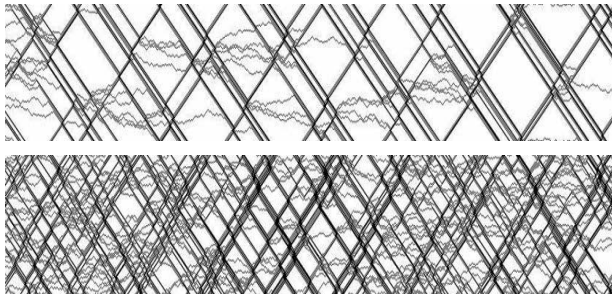
This section investigates the dynamic behavior of the system. Visual descriptions of the patterns that emerge from the interaction of the particles are given. Output patterns are analyzed in search for self-organization signatures. The



degree of clustering throughout the entire run is inspected, as well as the distance of the particle to their neighbors (measured in variation between fitness values).

The system was tested with stationary and time-varying populations. The experiments with static populations have been described by Fernandes et al. (2012); therefore, in this paper we only give an overview of the results and conclusions in that study in order to contextualize the discussion. The stationary model is described by the pseudo-code given in the previous section after removing step 3.

The main goals of this section are: 1) check if the self-organizing properties are maintained with time-varying fitness values; 2) investigate the properties of the dynamic and partially connected Bak-Sneppen model and compare it to the standard models.



**Figure 1.** Space-time diagrams of a 1-dimensional habitat.  $X \times Y$ :  $150 \times 1$ . Swarm: 25, 50 (top to bottom).

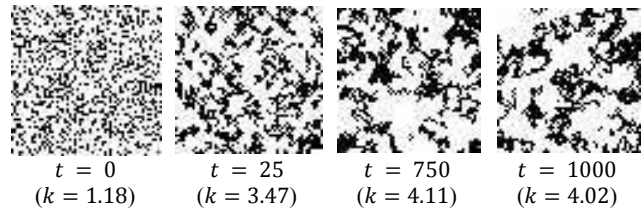
### Stationary Fitness Values

Although the model has been designed has a 2-dimensional framework for EAs, the 1-dimensional version may be constructed by setting  $X$  or  $Y$  to 1 (see the pseudo-code in the previous section). The 1-dimensional version displays interesting and complex behavior, has shown in Figure 1. The graphics represent the space-time diagrams of the system. These diagrams are usually used to track the spatial configuration of a cellular automaton over a number of time-steps. In this case, the diagrams may illustrate the chaotic and order factors of the system.

Results with grid size  $150 \times 1$  and  $n = 25$  and  $n = 50$  are shown in Figure 1. The leftmost row of the cells is the 1-dimensional lattice set up with a random initial distribution of particles. Each successive row going right is the updated lattice at the next time step. The diagrams show a mixture of order and randomness which is typical, for instance, of class 4 cellular automata. Some clusters of particles move up or down, while free particles randomly move through the grid until they are “captured” by a cluster. Meanwhile, clusters disaggregate, freeing more “wandering” particles. These are typical signatures of complexity and activity between order and randomness. If these traits emerge in a 1-dimensional environment, it is expected that, at least, a similar degree of complexity is present in the 2-dimensional system.

In order to investigate the 2-dimensional model, the grid was then set to  $60 \times 60$  and the swarm size to 1200 (meaning that the ratio between particles and nodes is 1:3).

Figure 2 depicts the distribution of the particles on the grid at different time-steps between  $t = 0$  and  $t = 1000$ . The



**Figure 2.** Position of particles and average degree of clustering  $k$ .  $X \times Y$ :  $60 \times 60$ ;  $n = 1200$ .

average degree  $k$  of clustering is given. This variable measures the number of particles in each particle’s Moore neighborhood. The average  $k$  is the degree value averaged over the entire population.

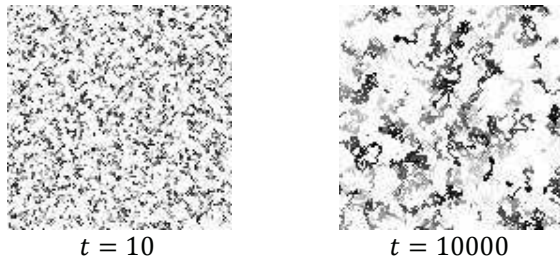
The images in Figure 2 show that the particles are able to self-organize into a dynamic structure of clusters and paths. This assumption is confirmed by the  $k$  values, which, starting from  $k = 1.18$ , tend to grow, reaching 4.02 after 1000 iterations. The graphics also confirm that the particles do not only aggregate in small clusters, they also form trails between the clusters. In fact, in most of the time-steps, large parts of the population are connected. This is a key result for the project of designing a dynamic self-organized framework for spatially structured EAs, since information may flow quickly through the population.

Another important outcome is observed in the snapshots of later iterations. Averaged  $k$  is similar at iterations 750 and 1000. In fact, at this later stage,  $k$  does not tend to increase. However, the distribution of the particles is clearly different in the two *snapshots* of the system. That is, even after converging to the maximum range of  $k$  values, the swarm continues to reorganize and reshape the clusters. The system is in a state of dynamic equilibrium. Clusters form, but they may disaggregate at any moment, and the particles move to another region of the habitat where they will cluster again with other particles.

Figure 3 shows the distribution of fitness values on the grid by plotting the particles with a grey-level proportional to their fitness. Comparing the distributions at an early and later stage we see that the particles do not only self-organize into clusters; they also tend to cluster according to the fitness, creating structures of particles with similar fitness.

A quantitative analysis of the system was conducted by investigating its output variables, namely the average degree of clustering  $k$  and the average distance  $d$  to the neighbors. The Fourier Transform of  $k$  and  $d$  was calculated for a representation of the signal in the frequency domain. For the Fourier Transform, 4096 samples of the signals were used, from  $t = 1000$  to  $t = 5095$ . This way, the spectral density leaves out the transient phase, from the random configuration at  $t = 0$  to the self-organized state. The observation and analysis of the spectral density showed that large regions of the spectra are reasonably approximated by power-laws.

The power spectra were plotted in log-log coordinates, as is customary, since the logarithmic transform renders the power spectrum a straight line whose slope can be easily estimated. The slope  $\alpha$  of the power-law in both cases was found to be close to 1, which is the slope of pink noise. The more general case, which displays a spectral density  $S(f) = \text{constant}/f^\alpha$ , where  $0 < \alpha \leq 2$ , is sometimes referred simply as  $1/f$  noise.



**Figure 3.** Distribution of fitness values on the grid. Lighter grey areas correspond to particles with lower fitness.  $X \times Y$ :  $60 \times 60$ ;  $n = 1200$ .

If we investigate the spectrum of  $k$  and  $d$  that emerges from a random structure, we find an almost flat density, a signature of white noise. The stigmergic rule supply the system with a typical trait of complex adaptive system and self-organization in near-equilibrium state between order and chaos.

Table 1 show the slopes of the power-laws used for fitting the data obtained by different ratios between the grid size and the number of particles. The relationship between intensity and frequency of  $k$  and  $d$  is similar when the ratio is in the range  $[1:24, 1:2]$ . Outside this range,  $\alpha$  tends to decrease. This is an expected result, due to the physical constraints of the system. On one hand, the swarm requires critical mass to interact. On the other hand, the particles require space to move. However, the model seems to be robust. In order to study its robustness, the swarm was tested with a fixed ratio between the population size and the number of nodes. Several combinations of  $n$  and grid size were used. The slopes of the power-laws used for fitting  $k$  and  $d$  spectrum are in Table 2. With  $n = 33$  the slope of the power-law decreases, but for  $n > 33$  the power-laws are very similar. The properties of the signals are stable for three orders of magnitude. The system is robust as long as the ratio is within a specific range. The complete description of these experiments, as well as other details on the results with the stationary version of the model, are given by Fernandes et al. (2012).

Table 1. Slope  $\alpha$  and  $r$ -squared of the power-law that fits the  $k$  and  $d$  spectral density for different ratios between  $n$  and the number of nodes on the grid ( $X \times Y$ ).

$n: \text{nodes} \rightarrow$	1:24	1:12	1:6	1:3	1:2	1:1.5	1:1.2
$k$	1.18 (0.76)	1.23 (0.76)	1.23 (0.76)	1.20 (0.76)	1.07 (0.70)	0.88 (0.60)	0.56 (0.60)
$d$	0.82 (0.60)	1.00 (0.72)	0.97 (0.68)	1.01 (0.69)	1.00 (0.69)	0.93 (0.64)	0.42 (0.60)

Table 2. Slope  $\alpha$  and  $r$ -squared.  $n: \text{nodes}$  is fixed and equal to 1:3.

$n \rightarrow$	33	75	147	300	616	1200	2408	4800
$k$	1.15 (0.72)	1.29 (0.77)	1.18 (0.75)	1.22 (0.77)	1.18 (0.74)	1.20 (0.76)	1.17 (0.74)	1.18 (0.76)
$d$	0.87 (0.62)	1.04 (0.70)	1.04 (0.71)	1.10 (0.75)	1.03 (0.70)	1.01 (0.69)	1.02 (0.69)	0.97 (0.69)

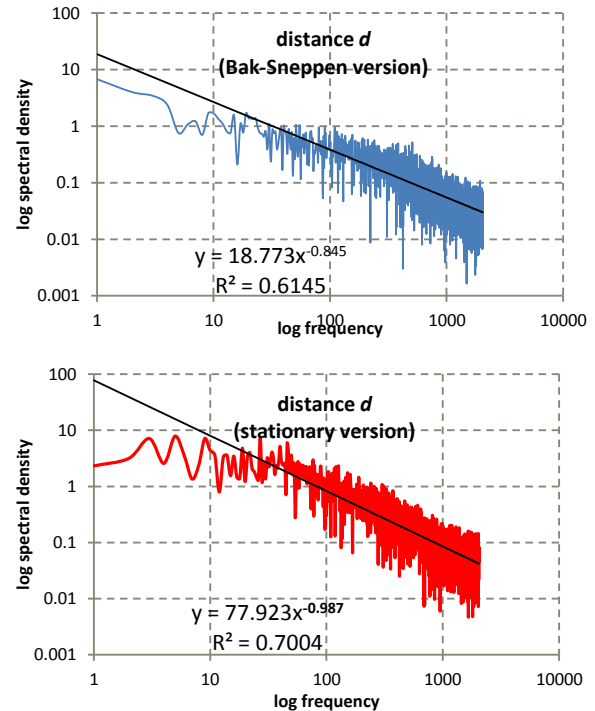
## Time-Varying Fitness Values

In this paper, the model was tested with the Bak-Sneppen mutation rules (i.e., including the step 3 of the pseudo-code given in the previous section). The size of the grid was set to  $60 \times 60$  and the swarm is comprised of 1200 individuals.

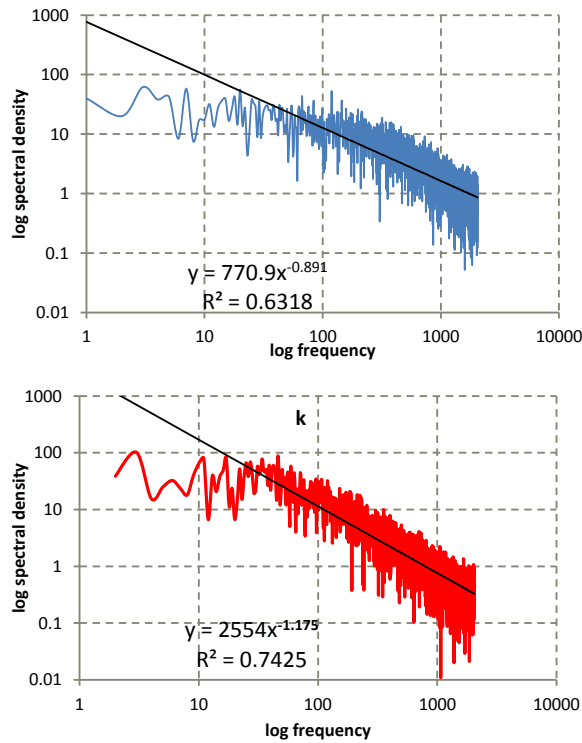
The first analyses aim at comparing the behavior of the system with stationary and non-stationary fitness values. For that purpose, the spectra of the output variables ( $k$  and  $d$ ) were computed and compared with the spectral densities of the stationary version.

Figure 4 compares the spectral density of the average distance between neighboring particles in each time-step. The introduction of the mutation rules based on the Bak-Sneppen model does not affect significantly the distribution of frequencies.

Figure 5 shows the spectral density of the connectivity degree  $k$ . Again, introducing a mutation mechanism in the original model does not affect the general behavior of the swarm and the clustering dynamics. These results demonstrate that it is possible to obtain an emergent behavior consisting of dynamic clustering based on similarity and hierarchy using not only a population of stationary fitness values, but also an evolving population. This is an important result since an EA, by definition, is a population of solutions that, in average, improves over time. If an EA is implemented on a population of the model, and if the intensity of changes is maintained within a certain boundary (here, the number of fitness values that change in each time-step is in the range  $[1,5]$ ), it is expected that global patterns that emerge from the proposed model also appear in the model-based EA.



**Figure 4.** Comparing the spectral density of the average distance  $d$  that emerges from the stationary and non-stationary fitness versions of the model.

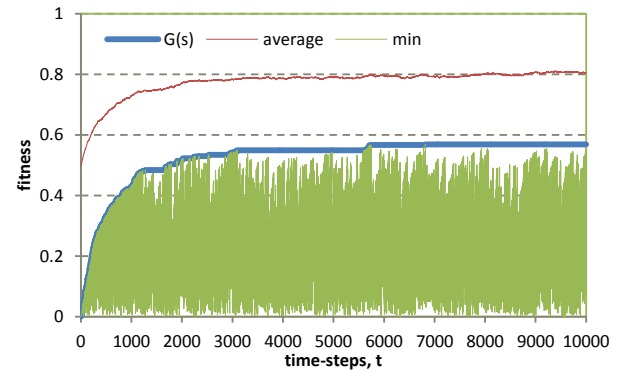


**Figure 5.** Comparing the spectral density of the average connectivity degree  $k$  that emerges from the stationary and non-stationary fitness versions of the model.

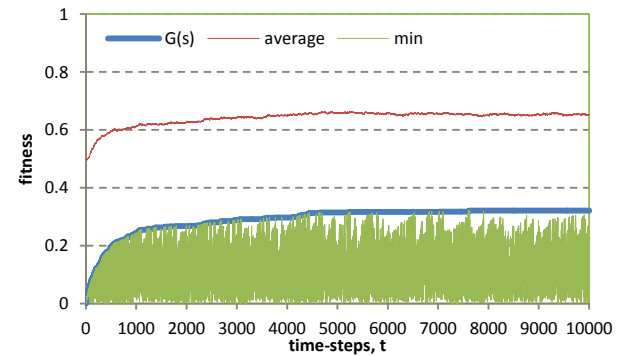
The evolution of the population can be visualized by plotting the average and minimum fitness of the population, as well as the gap function  $G(t)$ . Figure 6 shows the evolution of 1200 particles on a  $60 \times 60$  grid, while Figure 7 shows the evolution of a population of 1200 on  $30 \times 40$ , i.e., Figure 7 displays the behavior of a standard 2-dimensional Bak-Sneppen model. The average fitness of the partially connected model evolves to higher values. The proposed model reaches an average fitness values of approximately 0.8, while the standard 2D model stays below 0.7 (a result observed in several runs with different random seeds).

The gap function also grows faster and reaches higher values. In the several runs conducted for this study, the critical value of the gap function was found to be  $f_c \sim 0.6$ . The dynamics of the proposed model is clearly different from the standard 2D model. The sparser connection between the particles is a reasonable explanation for the differences in the evolutionary rates (please remember that in our model there are  $p \leq 1 + 2 \times d$  particles that are mutated in each time-step, while in the standard 2D Bak-Sneppen model there are  $p = 1 + 2 \times d$  mutations). The effects of the local movement rules are harder to measure, but since the particles cluster according to the fitness values, better particles tend to gather in the same regions, and therefore the mutation of the worst individuals will tend affect also *weak* neighbors, thus leading to a faster evolution of the population's fitness values.

One of the SOC signatures of the Bak-Sneppen model is the power-law relationship between the duration of the species' periods of stasis (time-steps between successive mutations) and their frequency. The proposed model displays



**Figure 6.** Evolution of 1200 particles on a  $60 \times 60$  grid. Average fitness, minimum fitness and gap function.



**Figure 7.** Evolution of 1200 particles on a  $30 \times 40$  grid (standard 2D Bak-Sneppen model).

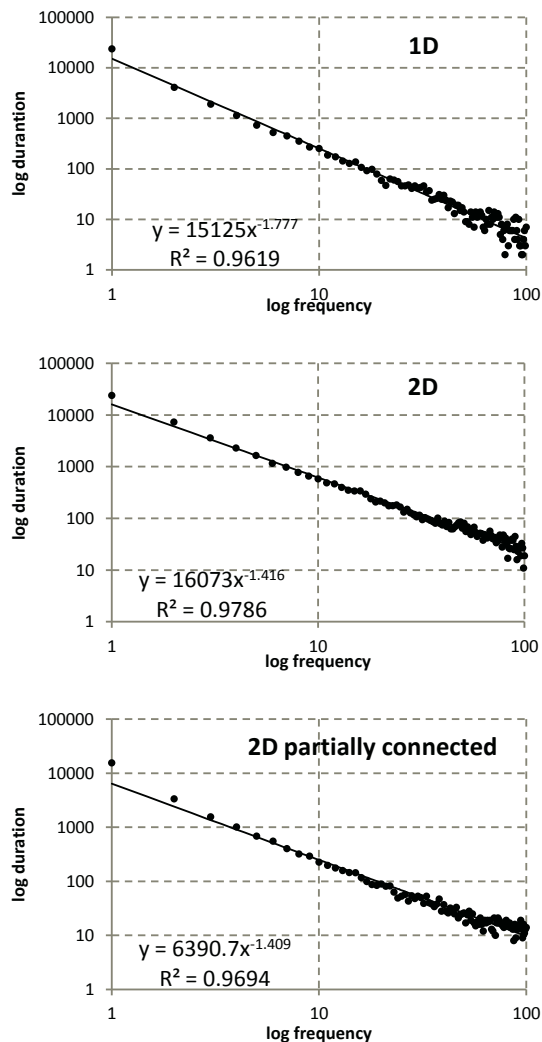
the same signature. The exponent of the power-law is approximately  $3/2$ , as seen in Figure 8. This is the same exponent obtained with the standard 2-dimensional model, while the 1-dimensional Bak-Sneppen system, in our experiments, displays a power-law with exponent approximately  $7/4$ .

The model maintains the characteristics of the stationary version proposed by Fernandes et al. (2012). Global patterns of clusters connected by paths tend to emerge. These clusters are highly dynamic, and in a few generations the distribution of the particles in the habitat may dramatically change (we believe there is an avalanche-based self-organized phenomenon behind the massive reconfigurations of the system but we haven't yet identified it). The output variables of the system display pink noise spectral densities. Furthermore, the proposed model maintains the characteristics of standard 2-dimensional Bak-Sneppen models. The average fitness of the population tends to grow with time, and the gap function converges to a specific critical value. The power-law observed in the distribution of distances between successive mutations also appears in the proposed model, with the same exponent as the 2-dimensional Bak-Sneppen model.

## Conclusions and Future Work

This paper describes an evolutionary extension of the self-organized swarm intelligence system proposed by Fernandes





**Figure 8.** Duration of the periods of stasis (periods in-between mutations).

et al. (2012). The system is a swarm of simple particles that interact on a heterogeneous grid of nodes. The particles communicate via the grid, and move according to simple rules. A fitness value is assigned to each particle. In each time-step, the fitness values of the worst particle and its neighbors are mutated. This is the basic rule of a Self-Organized Critically (SOC) model known as the Bak-Sneppen model of co-evolution between interacting species.

The system has been designed as a base-framework for spatially structured Evolutionary Algorithms (EAs). The original model (without the Bak-Sneppen mutation rules) displays a complex behavior illustrated by dynamic clustering of the particles, catastrophic reconfigurations of the distribution of the particles on the grid, and output variables with pink noise spectral densities. The model proposed in this paper maintains the main characteristic of the stationary fitness values version. This conclusion is very important for the project of designing a spatially structured framework for EAs based on the proposed system. Furthermore, the system

displays the same SOC signatures as the standard 2-dimensional Bak-Sneppen model.

In the future, the research will be focused on two main lines of work. Firstly, an EA will be implemented on the model and compared to standard spatially structured EAs. Secondly, the behavior of the system as an (hypothetical) SOC system will be studied. Traits such as the critical fitness threshold and the critical exponents of the model will be investigated. Furthermore, we believe that there is an avalanche-based phenomenon triggering the massive reconfigurations of the system (particles' positions on the grid). In a future research, we will try to identify that phenomenon, its origin, and study its distribution in search for self-organization signatures.

## Acknowledgments

The first author wishes to thank FCT, *Ministério da Ciência e Tecnologia*, his Research Fellowship SFRH/BPD/66876/2009, also supported by FCT (ISR/IST plurianual funding) through the PIDDAC Program funds. This work was supported by FCT PROJECT [Pest-OE/EEI/LA0009/2011], and also by project TIN2011-28627-C04-02, awarded by the Spanish Ministry of Science and Innovation, P08-TIC-03903 awarded by the Andalusian Regional Government, and CEI2013-P-14, awarded by the CEI-BioTIC UGR.

## References

- E.. Alba and B. Dorronsoro (2005). The exploration/exploitation tradeoff in dynamic cellular genetic algorithms, *IEEE Transactions on Evolutionary Computation* 9: 126–142.
- P. Bak, C. Tang and K. Wiesenfeld, Self-organized Criticality: an Explanation of 1/f Noise, *Physical Review Letters* 59(4), (1987), pp. 381-384.
- P. De Los Rios, M. Marsili, M. Vendruscolo, (1998), High dimensional Bak-Sneppen model, *Phys. Rev. Lett.* 80(26), 5746-5749.
- C.M. Fernandes, J.L.K. Laredo, J.J. Merelo, C. Cotta, A.C. Rosa, (2012) Towards a 2-dimensional Framework for Structured Population-based Metaheuristics, in *Proc. ICCS'12: IEEE International Conference on Complex Systems*, 1-6.
- P.-P. Grassé (1959) La reconstruction du nid et les coordinations interindividuelles chez bellicositermes et cubitermes sp. La théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs, *Insectes Sociaux*, 6 : 41-80.
- J. Kennedy and R.C. Eberhart. (2001) *Swarm Intelligence*, Morgan Kaufmann, San Francisco
- M. Tomassini (2005) *Spatially Structured Evolutionary Algorithms*, Springer, Heidelberg
- V. Gordon, L. Whitley (1993) Serial and Parallel Genetic Algorithms as Function Optimizers, in *Proc. 5th ICGA*, 177-183.
- J.L. Payne and M.J. Eppstein, (2006) Emergent mating topologies in spatially structured genetic algorithms, in *Proc. 8th GECCO*, 207-214.
- J.L.J. Laredo, A.E. Eiben, M. van Steen, J.J. Merelo, (2010) EvAg: a scalable peer-to-peer evolutionary algorithm, *Genetic Programming and Evolvable Machines* 11( 2): 227–246.
- R. Steinmetz and K. Wehrle, Eds. (2005) *Peer-to-Peer Systems and Applications*, Lecture Notes in Computer Science, vol. 3485, Springer.
- J.M. Whitacre, R.A. Sarker and Q. Pham (2008) The self-organization of interaction networks for nature-inspired optimization, *IEEE Transactions on Evolutionary Computation*, 12: 220–230.



# Evolving gene regulatory networks controlling foraging strategies of prey and predators in an artificial ecosystem

Joachim Erdei<sup>1</sup>, Michał Joachimczak<sup>2</sup> and Borys Wróbel<sup>2,3</sup>

<sup>1</sup>Department of Systems Modelling, Gdańsk University of Technology, Gdańsk, Poland

<sup>2</sup>Systems Modelling Laboratory, IO PAS, Sopot, Poland

<sup>3</sup>Evolving Systems Laboratory, Adam Mickiewicz University, Poznań, Poland

erdei@evosys.org, mjoach@evosys.org, wrobel@evosys.org

## Abstract

Co-evolution of predators and prey is an example of an evolutionary arms race, leading in nature to selective pressures in positive feedback. We introduce here an artificial life ecosystem in which such positive feedback can emerge. This ecosystem consists of a 2-dimensional liquid environment and animats controlled by evolving artificial gene regulatory networks encoded in linear genomes. The genes in the genome encode chemical products which regulate other genes, sense the environment (the scent of food, prey and predators), control the animat's movement, and its foraging strategy. An animat can switch multiple times in its life between two foraging strategies (with different metabolic costs): a predator can derive food from the prey, prey just from food that diffuses in the environment. When an animat consumes enough food (or prey), it produces an offspring with a mutated genome. Mutations introduce variation into the population, and this diversity together with selective pressures leads to the evolution of control for diverse foraging strategies in an ecosystem that can support hundreds of individuals.

## Introduction

The ability to prey on other organisms is a distinguishing feature of animals, and multi-level complex relationships between predators and prey are the building blocks of ecosystems. Prey-predator relationships create coupled selective pressures which can lead to evolutionary arms races between genes and lineages (species). Examples of artificial ecosystems in which such pressures exist include “sticky-feet” (Turk, 2010), in which simple multicellular organisms could both be prey and be preyed upon, and systems where separate lineages of prey and predators co-evolved, such as “Spiders” (Palmer and Chou, 2012), and “Bubbleworld” (Schmickl and Crailsheim, 2006).

In biology, prey-predator relationships evolve even between the simplest, one-celled organisms. The behaviour of these single cells is controlled by gene regulatory networks: networks in which a node represents a gene (or co-expressed genes) and edges represent regulation relationships—a directed edge from one node to another means that the product of one gene regulates the expression of another, often

because this product binds physically (thanks to chemical affinity to DNA) in the vicinity of another gene. Gene products play not only regulatory roles, but also catalyse chemical reactions in the cell, form intracellular or extracellular structures, including structures necessary to sense the changes in the environment or necessary to allow for cell movement.

In our previous work we evolved artificial gene regulatory networks—using a genetic algorithm, or a novelty search algorithm (Lehman and Stanley, 2011)—to process signals (Joachimczak and Wróbel, 2010), direct multicellular development (Joachimczak and Wróbel, 2008, 2012b), and control the behaviour of unicellular (Joachimczak and Wróbel, 2009) and multicellular animats (Joachimczak and Wróbel, 2012a). Other models of artificial gene regulatory networks have been evolved to match mathematical functions (Kuo et al., 2004), evolve biological clocks (Knabe et al., 2006), study dynamics of gene expressions (Reil, 1999), and evolve robot controllers (Reil, 1999; Quick et al., 2003), also using genetic algorithms and objective fitness functions. But a genetic algorithm or a novelty search algorithm is a very imperfect model of biological evolution. In biology there is no objective fitness function—fitness corresponds to the number of offspring that is produced, and can be construed as the ability to use the resources in the environment to do so.

Because the resources of the environment are always limited, the organisms compete for them, and offspring resembles parents but also varies, natural and artificial ecosystems can be analysed from the point of view of flows of energy/matter on a short time scale and from the point of view of information on how to use the resources (evolution) on the long times scale. Artificial ecosystems are very far from capturing the complexity of matter and energy transformations in Nature or the complexity of the evolutionary process (for a review, see Dorin et al., 2008). In this paper, we present a simple system in which artificial organisms (animats) obtain matter/energy from the environment and evolve. The animats metabolise food, producing waste. Matter and energy derived from food allows them to move and to produce offspring. Animats can sense the concentra-

tion of food and waste produced by other animats, and use this information to direct movement. Offspring receives a mutated genome from the parent, allowing for evolution.

The system we present here is an extension of our previous work, in which we coupled our Gene Regulatory evolving artificial Networks (GReaNs) artificial life system with a physically plausible model of a 2-dimensional liquid environment in which thousands of animats can evolve foraging behaviour (Erdei et al., 2012). The main contribution of the work here is the introduction of a simple metabolism and a model of animats that can switch their strategy from a predator to prey and vice versa. We show here a preliminary analysis of the evolutionary trajectories and of the evolved life strategies in our artificial ecosystem.

## 2-dimensional liquid environment with foraging animats and diffusible substances

Simulated organisms (animats) live and evolve in a toroidal 2-dimensional liquid environment. The world contains three diffusible substances: (i) food (a source of energy) diffuses from multiple points in the environment and also from killed prey, (ii) scent of prey, and (iii) scent of predators. The last two can be seen as waste products of the prey or predator cells and allow other cells to sense them. To allow for computationally efficient yet realistic simulation of diffusion, concentration of all substances is stored using a quadtree (Finkel and Bentley, 1974) in which the root node represents the whole environment, and the other nodes – subregions of this space. Each square subregion can be divided into four smaller and equal subregions, and thus each node can have exactly four children. The depth of the tree is higher for regions where either the concentration or the gradient of concentration is high (there is a separate quadtree for each substance). Places where animats are located are always represented by squares of minimum allowed size (1 length unit squared). In each subregion the chemical gradients are continuous, calculated using bilinear interpolation (Fig. 1, Gribbon and Bailey, 2004); to simplify this calculation, we permit only two kinds of neighbourhood: the neighbouring squares are of the same size or the bigger neighbour borders exactly 2 smaller squares, each one 4 times smaller in area than the bigger square.

Diffusion of substances between two adjacent squares follows the Fick's law:

$$\Delta P = \frac{c \cdot d}{D} \cdot (S_2 - S_1) \cdot \Delta t, \quad (1)$$

where  $\Delta P$  is the amount of food by which the concentration will increase in the next step in square 1 and decrease in square 2 (provided that the current concentration is greater in square 2 than square 1),  $c$  is the coefficient of diffusion (0.1 for food, and 0.25 for scent in the experiments described here),  $d$  is the length of the common edge,  $D$  is the distance between the centres of the squares,  $S_1$  and  $S_2$  are the current

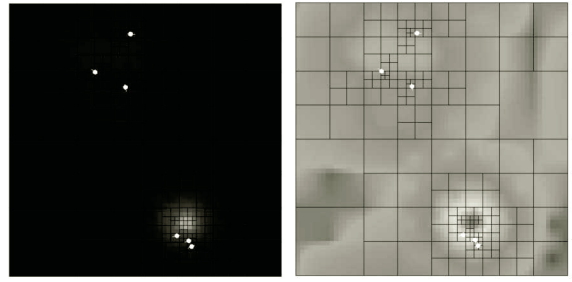


Figure 1: Modelling diffusion in 2-dimensional environment using a quadtree with continuous gradient of diffusible substances. The concentration is represented as a shade of grey, scaled linearly in the left panel and logarithmically in the right one; white corresponds to the maximum concentration close to the sources (bright squares). Animats (white circles) sense interpolated value of concentration at their exact location.

concentrations in both squares, and  $\Delta t$  is the duration of the simulation step. Since each square stores the concentration (not the amount), the concentration in square X is changed by  $\Delta S = \frac{\Delta P}{A_X}$ , where  $A_X$  is the area of square X.

At every time step, substances not only diffuse, but also degrade exponentially:

$$S(t_0 + \Delta t) = S(t_0) \cdot g^{\Delta t}, \quad (2)$$

where  $S(x)$  is the concentration of the substance in given square in time  $x$ ,  $\Delta t$  is the duration of the simulation step,  $g$  is the degradation coefficient (0.99 for food, 0.9 for scent), and  $t_0$  is the previous moment of time. Changes in concentration caused by diffusion and degradation make square areas split or merge, changing the quadtree.

## One-celled predators and prey controlled by gene regulatory networks

Each animat in our system has one cell; all have equal size, shape (a circle 1 length unit in diameter), two actuators and six sensors (Fig. 2). Each sensor provides the information on the concentration of one of the following substances: food, predator scent, and prey scent (a pair of sensors for each substance) at the sensor's location. An actuator generates thrust, and because we consider our animats as models of single cells, we use a word 'flagellum' when referring to actuators (but they can be thought of as thrusters or motor-driven wheels). The activity of each actuator is controlled in a continuous fashion, so the animats can go forward or rotate in the chosen direction by varying the level of activation of the actuators. The animats move faster (accelerate by  $\frac{\text{distanceUnit}}{\text{timeStep}^2}$ ) when both flagella are fully activated, or rotate faster (by  $\frac{1\text{rad}}{\text{timeStep}^2}$ ) when one flagellum is fully activated and the second one is not activated at all. The maximum

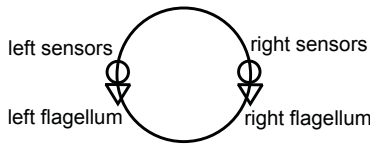


Figure 2: One-celled animat with six sensors (circles; each circle is the position of 3 sensors, one sensor for each substance in the system) and two actuators ('flagella'; triangles). The front of the animat is on the top of the figure.

speed is limited by drag (linear and angular) proportional to the velocity squared.

The concentration of food ( $S$ ) in the square where the centre of the animat is located determines how much food it consumes:  $0.75 \cdot S \cdot 1vu$ , where  $1vu$  is 1 volume unit. The food is stored internally and used up as follows:

$$\Delta M_t = (M_b + M_m \cdot \frac{a_1 + a_2}{2} + Z \cdot M_p) \cdot \Delta t, \quad (3)$$

where  $\Delta M_t$  is the total metabolic expenditure (by which the internal store is depleted in each time step),  $M_b$  is the base level metabolism (0.003 food units),  $M_m$  is the metabolic cost of movement (0.004),  $a_n$  is the current activation level of  $n^{th}$  flagellum (the minimal activation of a flagellum is 0, the maximum level is 1),  $Z$  is current state (prey have  $Z = 0$ , predators have  $Z = 1$ , and so does a prey cell undergoing a change into a predator; this change takes 20 simulation time steps), and  $M_p$  is metabolic cost of predation (in various experiments, we used 0.005, 0.010, or 0.015 for  $M_p$ ).

An animat can choose either to feed only on the food diffusing in the environment (and run the risk of being preyed upon) or to be a predator (and have a higher cost of metabolism, which can be seen as the cost of maintaining cellular structures necessary for killing the prey and feeding of other predators; in our system predators cannot feed on other predators). The switch between these two feeding strategies depends on the internal level of a chemical product encoded by the genome (Fig. 3). This concentration can change (in 0-1 range) during animat's lifetime, so multiple such strategy switches are possible (every time a threshold of 0.5 is passed).

The fact that each switch takes time (20 time steps) prevents the evolution of a strategy to change the state in response to prey or predators nearby. Depending on the state, an animat emits either a predator or prey scent. Only the collisions between predators and prey are detected, otherwise two animats in the same state (e.g., two prey cells) can overlap. Because the area of the world is large (in comparison to the average number of individuals), such overlaps happen rarely. If a prey cell is touched by a predator, the prey dies, provides 4 units of food to the predator, and whatever food was stored in its internal store to the square in which the prey cell was located.

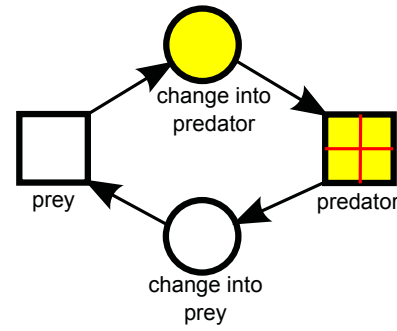


Figure 3: The possible changes between animat states. Depending on the state of its gene regulatory network, the state of the animat can change from prey (white square) to predator (yellow square; the red cross marks the state in which an organism can kill prey and cannot be killed) and vice versa, through temporary states (circles) that last 20 simulation steps. In a state marked by yellow the animat emits a predator scent, otherwise it emits prey scent.

The amount of food stored in the internal store determines if the animat is viable or can produce offspring. When the store drops to 0, the animat dies and 3 food units are released to the grid square occupied by the dead cell. When, on the other hand, there is more than 7 units of food in the store, an animat can produce one offspring cell. The cost of producing offspring is 4 units; the rest of the food in parent's store is divided equally between two cells. The new animat inherits the state of the gene regulatory network and its prey vs. predator status from the parent, it cannot change this status for the next 20 time steps to prevent a parent from immediately killing the offspring or vice versa.

Artificial gene regulatory networks that control animats' behaviour are encoded in linear genomes as described previously (Joachimczak and Wróbel, 2008). The network can be represented by a graph in which nodes correspond to chemical products in the system and the edges correspond to regulatory relations. All the products can have continuous concentrations (the minimum concentration is 0, the maximum is 1), with the exception of one special product whose level is always 1. This product serves the same role as the bias input in artificial neural network. There are 6 other input products; the concentration of these products depends on the activation of the sensors, and there are two products for each chemical substance diffusing in the environment (food, predator scent, prey scent). The concentration of one product ( $i_{dif}$ ) in each pair depends on the difference in the concentration of a substance sensed by the right sensor and the concentration sensed by the left, detecting even small gradients across the body:

$$i_{dif} = \frac{1}{7.5} \cdot \log_{10}(|s_{right} - s_{left}|) + 1, \quad (4)$$

where  $s_{right}$  and  $s_{left}$  are the concentrations of the substance detected by sensors. The concentration of the second special product ( $i_{avg}$ ) in each pair depends on the mean concentration sensed on the right and left:

$$i_{avg} = \frac{1}{7.5} \cdot \log_{10}\left(\frac{s_{right} + s_{left}}{2}\right) + 1, \quad (5)$$

Apart from 7 input products, there are 3 output products: 2 control directly the activity of the two flagella, one determines whether an animat is currently a predator or a prey.

The topology of the regulatory network is encoded in a linear genome, which is a list of genetic elements. Each element stores 4 numbers: a type of the element, a sign, and 2 coordinates. There are 4 possible types: regulatory, coding, input, and output. Coding elements and inputs define products that have affinity to regulatory elements. A series of regulatory elements followed by a series of coding elements is a regulatory unit. An output element is a regulatory unit by itself, as if it was a regulatory unit with one regulatory element (with the sign and coordinates of the output element) and a virtual coding region coding a product that does not have affinity to any regulatory elements. Regulatory units and inputs correspond to nodes in the graph that represents the networks; the edges are defined by affinities. The affinity between two elements is determined by their coordinates. Each element defines a point in a 2-dimensional abstract space (which has nothing to do with the 2-dimensional liquid environment in which the animats move). The affinity is maximum if two points overlap, and decreases with the Euclidean distance between points, reaching zero if the distance is more than 5. If  $K$  products have affinities to the  $J$  regulatory elements of a regulatory unit, the concentrations of all products belonging to this unit  $L_{\Omega}$  will change depending on the affinities (Euclidean distances,  $d_{k,i}$ ) and concentrations of the regulating products in the previous step ( $L_k$ ):

$$L_{\Omega} = \frac{2}{1 + e^{-(\sum_{j=1}^J \sum_{k=1}^K L_k (m_k \cdot m_i \cdot \frac{10-2d_{k,i}}{10d_{k,i}+1}))}} - 1, \quad (6)$$

While the concentrations of products change during animat's life, the topology of the network (the number of nodes and edges) does not. It only changes when an offspring is produced — the parent keeps the old genome, the offspring receives a mutated copy, so genomes that encode individuals who manage to reproduce are maintained in the population (similarly to the microbial genetic algorithm; Harvey, 2011). There are 2 types of mutations that change the offspring genome: simple mutations acting at the level of a single genetic element and complex mutations, acting at the level of the whole genome. There are 3 types of simple mutations, each can occur independently, with probability 0.01: change of type, change of sign, and change of coordinates (each coordinate is modified by a random value

from a normal distribution with  $\mu = 0$ ,  $\sigma^2 = 5$ ). Complex mutations—deletions and duplications—happen each with a probability of 0.002 per genome; the number of genetic elements removed or copied after a randomly chosen element is drawn from a geometric distribution (with a mean of 10).

## Evolution of foraging strategies in an artificial ecosystem

We started each evolutionary run from 1 000 animats with randomly generated genomes consisting of 10 regulatory units (with 1 regulatory and 1 coding element in each), 3 outputs, and 7 inputs. At the start of each run, the animats had random locations in the environment 256 length units across. Each evolutionary run continued for 3 000 000 time steps, and the amount of food sources decreased linearly in time from 64 (a number high enough for the animats with random genomes to survive) to 24 at the end (increasing the selection pressure). Each food source provided 0.2 food units per time step, starting from the initial 60 food units, except for the first 64 sources, which had between 1 and 60 food units initially (a number drawn from a uniform distribution), otherwise sources would be depleted periodically. We replaced depleted source with new ones at new random locations.

We have simulated evolution for 36 independent runs in total, 12 runs for 3 values of metabolic cost of predation ( $M_p$  in Eq. 3): 0.005, 0.010, and 0.015; we kept other parameters without change (Table 1). Our previous results for the situation without predation (Erdei et al., 2012) indicate that evolution of foraging is efficient. Is it equally so in the presence of predators? How the metabolic cost of predation influences the efficiency of animats' strategies? Do animats evolve to track prey or avoid predators?

When the metabolic cost of predation was 0.015, in all the runs there were animats at the end of the run, but the lower was the cost, the higher was the chance of a population dying out (Fig. 4) with decreasing food availability over time. In particular, half of the runs with the lowest cost died out around time step 200 000, so we removed these runs from further analysis, and we show the other results only up to step 2 000 000, when the populations started to die out for the intermediate cost (all the trends we discuss, however, continue beyond this time point with no qualitative changes).

The number of individuals stabilized each time the number of food sources decreased (Fig. 5b), suggesting that the animats adapt to the new environmental conditions. When we tested the animats with random genomes in the environment with 24 food sources, they were not able to survive, indicating evolution of efficient foraging.

For the runs with the highest metabolic cost of predation, the final number of predators was small, but temporarily—when the food availability was still high—the environment supported more predators than prey (Fig. 5a). On the other hand, when the cost was 0.005 it did not pay to be prey—



Table 1: Parameters of the environment and animal metabolism in evolutionary experiments

parameter	value
food degradation coefficient	0.99
food diffusion coefficient	0.1
scent degradation coefficient	0.9
scent diffusion coefficient	0.25
duration of state change	20 time steps
no state change after division	20 time steps
base level metabolism	0.003 food units
metabolic cost of movement	0.004 food units
metabolic cost of predation	0.005, 0.010 or 0.015 food units
metabolic cost of reproduction	4 food units
reproduction threshold	7 food units
world size	256 square length units
length of evolutionary runs	3 000 000 time steps
food sources	64 to 24, decreasing with time
initial size of food sources	60 food units
source depletion rate	0.2 food units per time step

the average amount of prey in 12 runs decreased almost to 0—and when the cost was 0.010, the amount of prey was temporarily driven almost to zero, before at average it stabilized at a low level (Fig. 5a). For the intermediate cost, however, the total number of animats was the smallest (Fig. 5b), and the average amount of food stored in the animats' internal stores (amount of eaten food minus metabolic costs) per time step was the lowest (Fig. 5c), suggesting that resources were channelled to tracking prey or to avoiding predators. The amount of food stored per step was the highest when the metabolic cost of predation was the highest; because the animats avoided changing to predators, they did not suffer this higher cost at all.

We analysed in detail the behaviour of several individuals from the final populations evolved under the intermediate

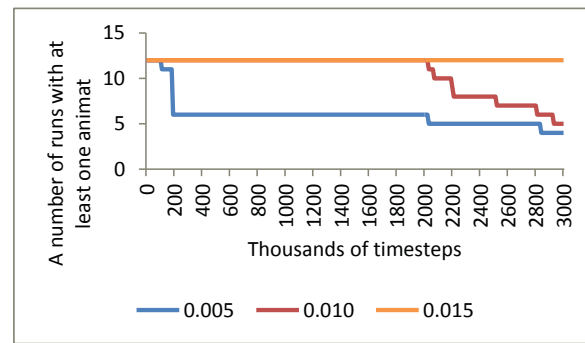


Figure 4: A number of evolutionary runs containing at least one animal as a function of time for three different values of the metabolic cost of predation.

metabolic cost of predation, by manual tracking of hand-picked individuals. This preliminary analysis suggests that a form of predator avoidance evolved in these conditions; although prey animats moved towards food sources, they avoided doing so when there were predators near the source. This strategy was efficient late in evolution because at this point prey was able to store enough food (in the internal store) to survive when searching for a food source without predators. If a prey individual moved, however, towards a source occupied by predators, it usually changed to a predator on the way. Although this change incurred a metabolic cost of producing defenses against other predators, it would pay out because high predator scent in a small area close to the source would not allow for efficient predator avoidance. On the other hand, we did not observe any predators that chased prey, even though prey did not move at full speed, and though it is easier to track another animal than to avoid it (because an animal can move faster than its scent diffuses, so the scent trail is left behind). Perhaps chasing did not evolve because the metabolic cost of movement increases linearly with speed, so a more efficient strategy is to consume food close to its source and to wait for prey there.

## Conclusions and future work

We present here an artificial life system in which animats evolved life strategies that involved searching for food, avoiding predators, and waiting for prey near food sources. The animats can sense the food diffusing from sources, and the prey or predator scent diffusing from other animats. We show that it is possible to simulate the evolution of hundreds of such animats using a simple, but still realistic model of a 2-dimensional liquid environment. In particular, we modelled diffusion using a grid (represented as a quad-tree) with the resolution that adapts dynamically to concentration of chemicals and the movement of animats. The animats move in continuous space, and we approximated continuous concentration gradients using bilinear interpolation.

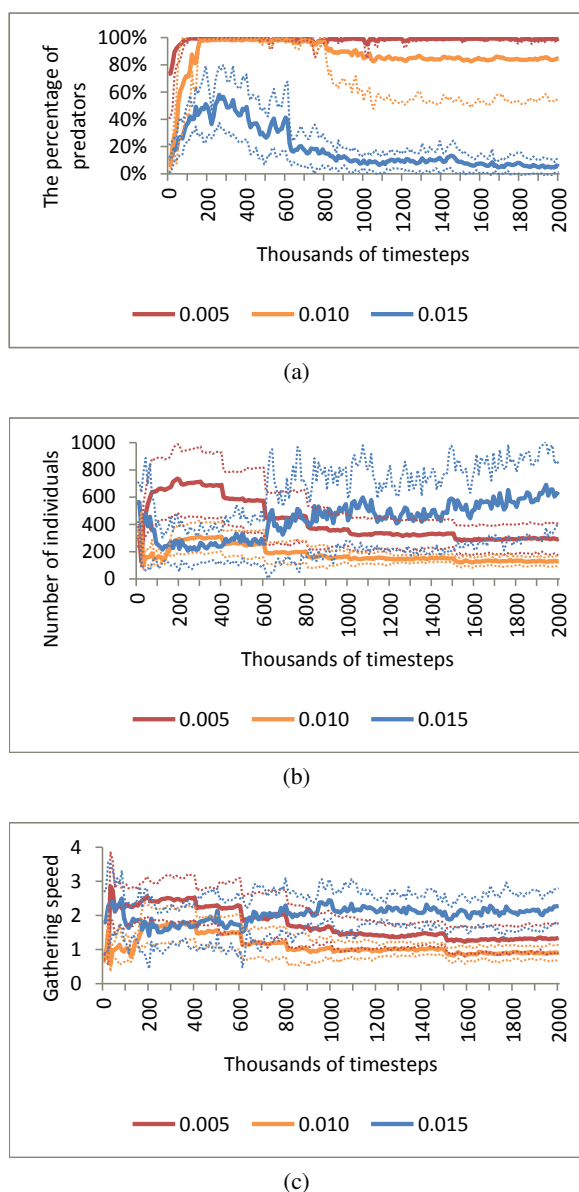


Figure 5: The percentage of predators, population size, and the amount of food stored by all the animats in the population per time step (gathering speed) for various metabolic costs of predation. Solid lines: averages over independent runs in which the populations did not die out before time step 2 000 000; dotted lines: averages  $\pm$  standard error.

The environmental conditions in our system can be adjusted to be qualitatively similar to those experienced by unicellular organisms. Some of such organisms—like the animats in our system—propel themselves using flagella and sense gradients across their one-celled bodies. Although many such organism live in 3-dimensional liquid environments, 2-dimensional environments (surfaces) also abound

in nature, and gravitation or other forces may deliver food to such surfaces in the form of food particles. The behaviour of our animats is controlled by gene regulatory networks, and the state of this network determines if an animat is a predator (this incurs higher cost) or prey. The cost of predation can be seen as the cost of producing cellular structures necessary to kill or digest the prey, and to defend the cell against other predators. The gene regulatory networks evolve in a way that is biologically realistic, without any objective fitness function or a genetic algorithm. The survival and reproduction in our system depends on animats' ability to find food and possibly prey (in the case of predators) or avoiding predators (in the case of prey). We plan to see in our future work if more complex environments (for example, obstacles, patchiness or seasonality of the food supply) or other environmental conditions will allow in our system for the evolution of other complex behavioural strategies and to the observation of general patterns in evolution (Dorin et al., 2008) in this virtual ecosystem.

## References

- Dorin, A., Korb, K., and Grimm, V. (2008). Artificial-life ecosystems: what are they and what could they become? In *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 173–180. MIT Press, Cambridge, MA.
- Erdei, J., Joachimczak, M., and Wróbel, B. (2012). Evolution of chemotaxis in single-cell artificial organisms. In *Proceedings of ICT Young 2012*, pages 185–190. Gdansk University of Technology, Poland.
- Finkel, R. A. and Bentley, J. L. (1974). Quad trees a data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9.
- Gribbon, K. T. and Bailey, D. G. (2004). A novel approach to real-time bilinear interpolation. In *Proceedings of the 2nd IEEE International Workshop on Electronic Design, Test and Applications, DELTA 2004*, pages 126–134. IEEE Computer Society.
- Harvey, I. (2011). The microbial genetic algorithm. In *Advances in Artificial Life: Proceedings of the 10th European Conference on Artificial Life, ECAL 2009*, volume 5778 of *Lecture Notes in Computer Science*, pages 126–133. Springer, Berlin / Heidelberg, Germany.
- Joachimczak, M. and Wróbel, B. (2008). Evo-devo *in silico*: a model of a gene network regulating multicellular development in 3D space with artificial physics. In *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 297–304. MIT Press, Cambridge, MA.
- Joachimczak, M. and Wróbel, B. (2009). Evolving gene regulatory networks for real time control of foraging behaviours. In *Artificial Life XII: Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pages 348–355. MIT Press, Cambridge, MA.
- Joachimczak, M. and Wróbel, B. (2010). Processing signals with evolving artificial gene regulatory networks. In *Artificial Life*

- XII: *Proceedings of the Twelfth International Conference on the Simulation and Synthesis of Living Systems*, pages 203–210. MIT Press, Cambridge, MA.
- Joachimczak, M. and Wróbel, B. (2012a). Co-evolution of morphology and control of soft-bodied multicellular animats. In *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation, GECCO '12*, pages 561–568. ACM, New York, NY.
- Joachimczak, M. and Wróbel, B. (2012b). Open ended evolution of 3d multicellular development controlled by gene regulatory networks. In *Artificial Life XIII: Proceedings of the Thirteenth International Conference on the Simulation and Synthesis of Living Systems*, pages 67–74. MIT Press, Cambridge, MA.
- Knabe, J. F., Nehaniv, C. L., Schilstra, M. J., and Quick, T. (2006). Evolving biological clocks using genetic regulatory networks. In *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 15–21. MIT Press, Cambridge, MA.
- Kuo, D. P., Leier, A., and Banzhaf, W. (2004). Evolving dynamics in an artificial regulatory network model. In *Proceedings of Parallel Problem Solving from Nature, PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 571–580. Springer, Berlin / Heidelberg, Germany.
- Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223.
- Palmer, M. E. and Chou, A. K. (2012). An artificial visual cortex drives behavioral evolution in co-evolved predator and prey robots. In *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference Companion*, pages 361–364. ACM, New York, NY.
- Quick, T., Nehaniv, C. L., Dautenhahn, K., and Roberts, G. (2003). Evolving embodied genetic regulatory network-driven control systems. In *Advances in Artificial Life: Proceedings of the Seventh European Conference on Artificial Life, ECAL 2003*, volume 2801 of *Lecture Notes in Computer Science*, pages 266–277.
- Reil, T. (1999). Dynamics of gene expression in an artificial genome implications for biological and artificial ontogeny. In *Proceedings of the 5th European Conference on Artificial Life, ECAL 1999*, volume 1674 of *Lecture Notes in Artificial Intelligence*, pages 457–466. Springer, Berlin / Heidelberg, Germany.
- Schmickl, T. and Crailsheim, K. (2006). Bubbleworld.evo: Artificial evolution of behavioral decisions in a simulated predator-prey ecosystem. In *From Animals to Animats 9: Proceedings of the 9th International Conference on Simulation of Adaptive Behavior, SAB 2006*, volume 4095 of *Lecture Notes in Computer Science*, pages 594–605. Springer, Berlin / Heidelberg, Germany.
- Turk, G. (2010). Sticky feet: evolution in a multi-creature physical simulation. In *Artificial Life XII: Proceedings of the Twelfth International Conference on the Simulation and Synthesis of Living Systems*, pages 496–503. MIT Press, Cambridge, MA.

# FARSA<sup>1</sup>: An Open Software Tool for Embodied Cognitive Science

Gianluca Massera<sup>2</sup>, Tomassino Ferrauto<sup>2</sup>, Onofrio Gigliotta<sup>3</sup>, Stefano Nolfi<sup>2</sup>

<sup>1</sup><http://laral.istc.cnr.it/farsa>

<sup>2</sup>Institute of Cognitive Sciences and Technologies, CNR, Rome, Italy

<sup>3</sup>University of Naples Federico II, Naples, Italy

{gmassera, tomassino.ferrauto, s.nolfi}@istc.cnr.it, onofrio.gigliotta@unina.it

## Abstract

In this paper we introduce the FARSA open-source tool that supports the accomplishment of experimental research in Embodied Cognitive Science and Adaptive Behavior. The tool provides a set of integrated libraries and a graphical interface that enable to design, to accurately simulate and to analyze individual and/or collective embodied robotic models. The modular architecture of the tool allows to progressively expand it with new software components and simplifies the implementation of custom experiments. The tool comes with a set of exemplificative experiments and with a synthetic but comprehensive documentation that should enable users to quickly master its usage.

## Introduction

The realization of the importance of embodiment and situatedness for the study of behavior and cognition led to a paradigm shift toward the so-called Embodied Cognitive Science (Clark, 1999; Pfeifer and Bongard, 2007; Shapiro, 2007). From a methodological point of view this change implies that models of behavioral and cognitive capacities should take into consideration the characteristics of the agent's nervous system, of the agent's body, of the environment and of the properties that arise from the agents/environment interactions. This in turn requires the formulation of models that are far more complex than their previous disembodied counterpart and that are not constituted simply by static descriptions but rather by processes that run in the physical world or in realistic computer simulations.

The recent development of robotic platforms that are relatively affordable and easy to use (such as the Khepera<sup>1</sup> and the Nao<sup>2</sup> robot) as well as the development of software libraries that enable the realization of realistic simulations of physical processes (such as ODE (Smith, 2004) and Newton Dynamics (Jerez and Suero, 2004)) constitute important facilitators for the design of embodied models. Despite of that,

the knowledge barrier that Embodied Cognitive Science researchers should face to build and analyze their models is still very high.

To mitigate this problem we developed FARSA, an open-source software tool that enables researchers and students to easily and effectively carry on research in Embodied Cognitive Science. FARSA combines in a single framework the following features:

- it is open-source, so it can be freely modified, used and extended by the research community;
- it is constituted by a series of integrated libraries that allow to easily design the different components of an embodied model (i.e. the agents' body and sensory-motor system, the agents' control systems, and the ecological niche in which the agents operate) and that allow to simulate accurately and efficiently the interactions between the agent and the environment;
- it comes with a rich graphical interface that facilitates the visualization and analysis of the elements forming the embodied model and of the behavioral and cognitive processes originating from the agent/environment interactions;
- it is based on a highly modular software architecture that enables a progressive expansion of the tool features and simplifies the implementation of new experiments and of new software components;
- it is multi-platform, i.e. it can be compiled and used on Linux, Windows, and Mac OS X operating systems;
- it comes with a set of exemplificative experiments and with a synthetic but comprehensive documentation that should enable users to quickly master the tool usage.

In section 2 we discuss the relation with other similar tools. In section 3 we review the main features and capabilities of the tool. In section 4 we describe the design and working principles of the software architecture. Finally in section 5, we describe the planned future extensions of the tool.

<sup>1</sup><http://www.k-team.com/mobile-robotics-products/khepera-ii>

<sup>2</sup><http://www.aldebaran-robotics.com/en/>



## Related Tools

Objectives similar to those we have tried to reach with FARSA have been actively pursued during the last 20 years by academic research laboratories, small private companies, and multinational companies. Here we restrict our analysis to the most related attempts.

One of the first and most influential tool is Webots™ (Michel, 2004) a mobile robot simulator, initially developed by Olivier Michel at the Swiss Federal Institute of Technology (EPFL) in Lausanne, Switzerland, and then commercialized by a small spin-off company led by the software creator. The tool is used by about 1000 research centers and universities worldwide. Webots™ includes a robot simulator with a rich collection of predefined robotic models, a visualization tool that allows to observe the robot's behavior, a library of methods for customizing the robot and the environment, and a library of simulated sensors and actuators. A limitation of this tool is constituted by its commercial nature that introduces a cost barrier, prevents the possibility to fully inspect and customize the source code, and limits the possibilities to exploit collaborative development.

ARGOS (Pinciroli et al., 2012) is a recently developed 3D physic simulation tool targeted particularly toward swarm robotics research. It is an open source project. The usage of the tool, however, require a significant programming effort also due to the lack of an integrated graphical interface.

USARSim (Carpin et al., 2007) is also open-source simulator that was initially targeted toward urban search and rescue scenarios and later extended toward a more general use. It supports a wide range of robotics platforms (humanoids, wheeled, vehicles, etc) and has been adopted as a simulation platform by the Robotcup initiative. USARSim is an open source project which, however, is based on the Unreal Engine proprietary technology. This limits the inspection and the customization of the tool at the level of the robot implementation and impose the use of a particular proprietary language (unreal script) for the configuration of the experiments.

Gazebo (Koenig and Howard, 2004) is another open-source simulator analogous to USARSim but does not use any third party proprietary code. Another advantage of Gazebo is that it can be used in combination with Player (Gerkey et al., 2001) a tool that can be used to design the agents' controller and eventually the adapting process. The two tools are constituted by independent software programs that communicate through a dedicated network protocol. The combination of these tools constitute a powerful environment. Its use, however, requires a significant programming expertise and learning efforts.

Finally, Microsoft and Willow Garage developed two similar robotic suites: Microsoft Robotics Developer Studio

(RDS)<sup>3</sup> and Robot Operating System (ROS)<sup>4</sup>. These packages, that constitute a sort of operating system and a developmental environment for robotics applications, include device drivers, libraries, visualizers, message-passing, package management, 3D simulation tools, visual programming languages, a library of simulated robotic platforms, sensors, and actuators. The limits of these tools, for what concerns the objective addressed in this paper, is their complexity and consequently the required expertise and learning efforts. Another technical limitation is constituted by the fact that they are not multi-platform: Microsoft RDS runs only on windows while ROS runs only on Linux, limiting community collaborations.

The FARSA project aims to provide an open-source and multi-platform tool that it is easy to use and to extend and, in addition to a robotic simulation environment, provides integrated tools for designing the control systems of the robots, for analysing the robots' behaviour, and for subjecting the robots to evolutionary and/or learning processes.

## Features and Capabilities

FARSA is a re-engineered and extended version of a tool that has been developed since the 1995 by Stefano Nolfi and Onofrio Gigliotta (Nolfi, 2000; Nolfi and Gigliotta, 2010) which has been used for research and education purposes by more than 50 research laboratories and universities. It is in an open-source software tool that can be freely used and modified and a cross-platform application that runs on Linux, Windows and Mac OS X (on either 32bit or 64bit systems). The tool can be downloaded from <http://laral.istc.cnr.it/farsa>. FARSA is well documented, easy to use and comes with a series of exemplificative experiments that allow users to quickly gain a comprehension of the tool. These experiments can be used as a base for running a large spectrum of new experiments that can be set up simply by editing a configuration file.

The tool is constituted by a series of integrated software libraries providing the features described in the following sub-sections.

### The Robots/Environment Simulator

The robots/environment simulator (*worldsim*) is a library that allows to simulate the robot/s and the environment in which it/they operate. The library supports both individual robot simulation and collective experiments in which several robots are placed in the same environment. The physical and dynamical aspects of the robots and of the robots/environment interactions can be simulated accurately by using a 3D dynamics physics simulator or by using a faster but simplified kinematic engine. For what concern the dynamics simulation, FARSA relies on the Newton Game

<sup>3</sup><http://www.microsoft.com/robotics/>

<sup>4</sup><http://www.willowgarage.com/pages/software/ros-platform>

Dynamics engine (Jerez and Suero, 2004) that enables accurate and fast simulations. The underlying dynamic engine has been encapsulated so to enable the inclusion of alternative engines.

Currently, FARSA supports the following robotic platforms: the Khepera (Mondada et al., 1994), the e-Puck (Mondada et al., 2009), the marXbot (Bonani et al., 2010) (see Figure 1, bottom) and the iCub (Sandini et al., 2004) (see Figure 1, top). These robots have been designed by assembling a series of building blocks (physical elements, sensors, and motorized joints) that users can re-use to implement alternative, not yet supported, robots.

In the case of the iCub, the simulator is based on the YARP (Metta et al., 2006) middleware library (the same command used to read the robot's sensors and control the robot's motor can be used to work with the simulated or real robot). This strongly facilitates the possibility to port results from simulation to reality and the possibility to integrate into FARSA projects the software modules available from the iCub software repository<sup>5</sup>. With respect to the iCub simulator developed by Tikhonoff et al. (2008), the simulation library included in FARSA presents a series of advantages: it strictly conforms to the real kinematic joints structure of the robot, it allow to simulate multiple robots, it includes both a dynamic and kinematic engine, and it provides an enhanced visualization tool.

### The Sensor and Motor Library

FARSA also includes a library of ready-to-use sensors and motors. In some cases, sensors and motors include software routines that pre-elaborate sensory or motor information (e.g. to reduce its dimensionality) and/or integrate different kinds of sensory-motor information (as in the case of motors that set the torque to be produced by a joint motor on the basis of the current and desired position of the controlled joint).

Wheeled robots are provided with infrared, ground, traction force, linear vision, and communication sensors, among others. Moreover, they are provided with wheels, grippers, LEDs, and communication actuators.

The iCub robot is provided with proprioceptors that measure the current angular position of the robot's joints, tactile sensors, and vision sensors among others and with actuators that control all the available DOFs.

The state of the robot's sensors and motors, as well as the state of selected variables of the robot's control system, can be graphically visualized while the robot interacts with the environment (see Figure 2). This provides an useful analysis and debugging tool.

### The Controller Libraries

These libraries enable the user to design, modify and visualize the robot's control system. Currently FARSA includes

<sup>5</sup>[http://wiki.icub.org/iCub\\_documentation/](http://wiki.icub.org/iCub_documentation/)

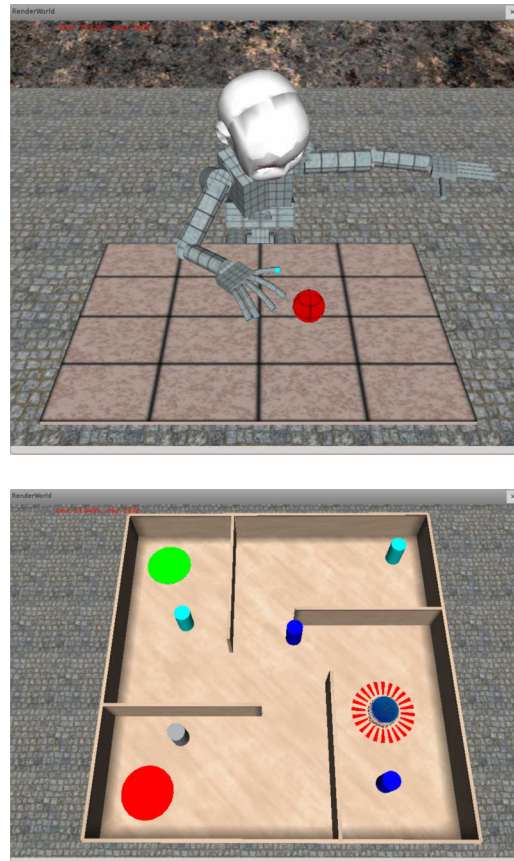


Figure 1: Snapshots taken from the 3D robot/environment renderer of FARSA. Top: A simulated iCub robot that reaches and grasps a spherical object located over a table. Bottom: A simulated marXbot robot that navigates in a structured environment containing walls and coloured objects.

two libraries that support the design of neuro-controllers. Users willing to use other architectures or formalisms can integrate into FARSA alternative libraries (see the section “Extending FARSA”).

Evonet is an easy-to-use library that enables users to graphically design, modify and visualize the architecture of the robot’s neural controller as well as the properties of the neurons and of the connection weights (see Figure 2). The library supports logistic, leaky integrator, and threshold neurons. NFW is an alternative object-oriented library that provides a larger variety of neuron types and output functions (Gaussian, winner-take-all, ramp, periodic, etc.) and supports the use of radial basis function neural network.

Thanks to the integration between the controller and the sensory and motor libraries, the sensory and motor layer of the neural controller is automatically generated on the basis of the selected sensors and motors. Moreover, the update of the sensory neurons and the update of the actuators on the basis of the state of the motor neurons is handled automatically.

Finally, the graphic viewer of the robot’s controller (see Figure 2) also enables users to lesion and/or to manually manipulate the state of the sensors, internal, and motor neurons in order to analyze the relationship between the state of the controller and the behavior that originates from the robot/environment interaction.

## The Adaptation Libraries

These libraries enable the user to subject a robot or a population of robots to an adapting process (i.e. to a evolutionary and/or learning process during which the characteristics of the robots are varied and variations are selected so to improve the abilities of the robots to cope with a given task/environment).

The adaptation libraries that are currently available support the use of evolutionary algorithms (including steady state, truncation selection, and Pareto-front algorithms), supervised learning algorithms (i.e. back-propagation), and unsupervised learning algorithm (i.e. Hebbian learning). The evolutionary algorithms are parallelized at the level of the individual’s evaluation and can therefore run significantly faster in multi-core machines and computer clusters.

In the case of evolutionary and supervised algorithm, the variation in performance during the adaptation can be monitored and analyzed in the associated graphic renderer (see Figure 3).

## Design and Working Principles

The architecture of FARSA is based on three key ideas: the *components*, the *configuration file* and the *plugins*.

The *components* are software modules that implement a given object or process. They can be organized in a hierarchical manner. For example, a project might include an

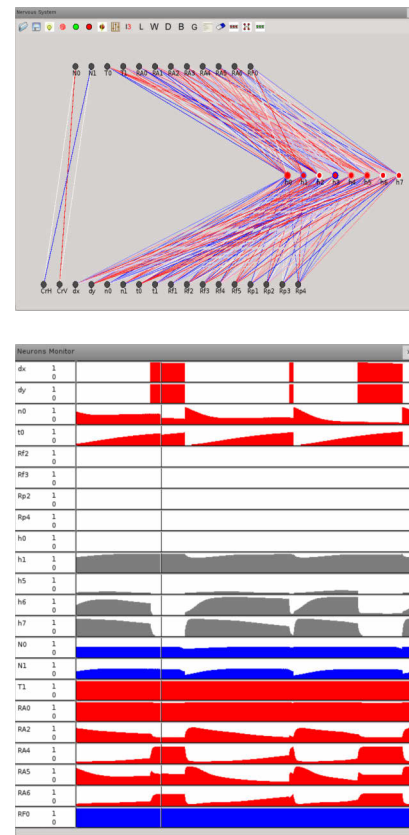


Figure 2: Top: The controller graphic widget that allows to visualize, modify, and analyze the robot’s neural architecture, the strenght of the connection weights and biases, and the properties of the neurons. Bottom: The controller monitor that displays the activation state of the sensory, internal, and motor neurons while the robot interacts with the environment.

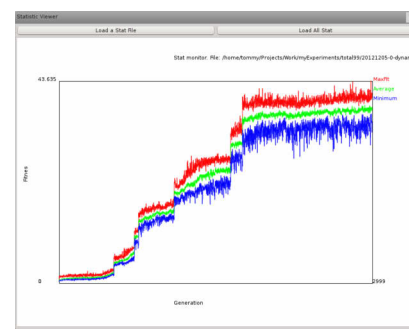


Figure 3: The graphic widget of the adapting process. In this example, the widget is used to show the best, average and worst fitness of an evolutionary experiment through out generations.



evolutionary process component, that includes as subcomponent an experimental component, that includes as subcomponent an iCub robot component, a neural network controller component, and several sensors and motors components. The main characteristic of components is that they can be automatically instantiated and configured from the content of a *configuration file* (they have a direct relation to groups of parameters in a *configuration file*, as explained below). Components might also include associated commands (e.g. <“evolve”, “stop”, “test”> in the case of an evolutionary component), and graphical widgets that can be accessed by the FARSA main graphic interface (see next section).

The *configuration file* is a text file that specifies the components (e.g. the robotic platform, the robots’ sensors and the motors, the robots’ controllers, and eventually the robots’ adapting process) that are going to be used in a particular project and the parameters (e.g. the number of robots situated in the same environment, the number and type of objects present in the environment) that are used to configure them. The file has a hierarchical structure analogous to the hierarchical organization of components. The configuration file is a human readable text file (in .INI or .XML format) that can be edited through the Total99 graphic interface (described in the next section) or directly through a standard text editor. This enables users to configure and run experiments on remote machine (e.g. computer clusters) that do not have a graphical environment. The modular and hierarchical organization of components combined with the configuration file has several advantages:

- it allows to instantiate at runtime only the components that are needed in a particular project, thus eliminating the risk that problems affecting other components might affect the functionality of the whole project,
- it gives the possibility to re-use the same components in different projects,
- it enables a progressive expansion of the tool with the development of additional components,
- it simplifies the tool usage through the visualization of only the parameters, the commands, and the graphic widgets that are relevant for a given project.

A *plugin* contains compiled code of new components or features created by users. They might consist of subclasses of existing components (e.g. a subclass of an evolutionary experiment with a new implemented fitness function or a new subclass of the sensor class implementing a new type of sensor not available in the sensor library) or of completely new components (e.g. a behaviour-based controller tool with associated parameters, commands and graphic widgets). The plugins, which are loaded and instantiated at run time, are totally equivalent to the other native components of FARSA for what concern the functionalities and use (e.g. they can be

configured and commanded in the same manner and through the same graphic interface of the native components). Plugins provide several advantages:

- they enable users to neatly separate their new code from the main library,
- they facilitate the distribution and sharing of additional components and feature within the FARSA community,
- they enable users to get access to a number of exemplary experiments that increase over time,
- they allow authors of scientific papers to provide an easy way to replicate their work.

Overall the workflow in FARSA is as follows: the project configuration file and the required plugins are loaded, the required components are created and configured on the basis of the configuration parameters, the associated commands and graphical widgets are created and made available to the user through the graphic interface.

### The graphical interface

Total99 is the graphical interface that allows to configure experiments, to instantiate the required software components, and to use the associated commands and graphic widgets. Total99 can also operate in batch mode without graphics if required. Total99 can be used to create, view, or modify a configuration file (Figure 4). This can be done by loading or creating a configuration file (through the use of the commands available in the File menu) and by then setting the configuration components and parameters through the parameters widget (orange rectangle of Figure 4).

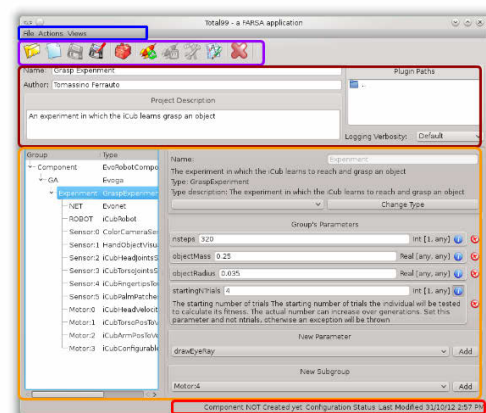


Figure 4: The Total99 graphical interface. The menu bar (blue), the toolbar (magenta), the project information bar (brown), the project parameters widget (orange), and the status bar (red) have been highlighted with coloured rectangles.



More specifically, the left part of the parameters widget is used to display the hierarchical organization of the components and the right part is used to display the parameters of the currently selected component and/or to add or remove sub-components and parameters (these can be selected from automatically generated lists that include only the parameters that belong to the current component and the sub-components that can be instantiated from the current component).

Once the configuration file has been set up, the user can run the project through the menu or the tool bar. As we mentioned above, this initiates the loading of the selected plugins, the instantiation of the software components specified in the configuration file, and the configuration of the components on the basis of the parameters specified in the configuration file. At this point, the commands associated to the components that have been instantiated and the associated graphic widgets can be executed from the Action and Views folders of the menu bar.

### Extending FARSa

FARSa can be extended by implementing additional software components. The integration of the new components into FARSa, that enables the possibility to use, execute, and configure them as native components, requires to fulfill the following three requirements.

The first requirement is that the class of the new component should be defined as a sub-class of an existing component or of a virtual empty component.

The second requirement is that the new class should include a `describe()`, `save()`, and `configure()`<sup>6</sup> functions that are used respectively for declaring the properties of the configuration parameters, saving the current value of the parameters on the configuration file when requested, and configuring the object on the basis of the parameters. For all these operations, FARSa provides helper functions in order to simplify and to minimize the effort of writing them.

The third requirement is that the new component has to be registered with the `registerClass()` function specifying the component type name.

In the case of components that also include new commands and/or new graphic widgets (that should be made available from the Total99 graphic interface), the user should also declare the new commands and/or widgets by implementing the `fillActionsMenus()`, `getViewers()`, and `addAdditionalMenus()` functions.

More detailed information can be found in the on-line documentation.

### Future Plans

The first stable version of FARSa has been just released online. During the next few months we plan to refine and

<sup>6</sup>Instead of implementing the `configure()` function, it is possible to implement a special constructor of the class

extend the documentation and the library of the exemplificative experiments. Then we plan to keep extending the tool, to promote the development of a community of users and developers, and to develop multimedia materials that can enrich the educational and training potential of the tool.

### Planned Extensions

Currently planned extensions include: (i) additional controller and adaptive software components that will enable the direct use of other formalisms and training algorithms (e.g. self-organizing maps, fuzzy networks, reinforcement learning) as well as the combined use of different techniques (e.g. unsupervised and supervised learning algorithms), (ii) additional readily available robotic platforms, and (iii) the possibility of using a simple script language as an alternative to C++ to implement new types of experiments.

### FARSa Community

The full open-source nature of FARSa and its modular architecture constitute two important prerequisites for the development of a wide community of users and developers that can profit from the tools and can contribute to its further extension. To enable the formation of such community we plan to widely disseminate the tool and to create web-based facilities that can be used to store users' knowledge and plugins.

### Educational Use of FARSa

The simplicity of use potentially enables FARSa to also become an excellent educational tool. To promote this use we plan to develop and promote the collaborative development of tutorials and training material targeted toward undergraduate and graduate courses in Embodied Cognitive Sciences and Autonomous Robotics. Finally we plan to develop examples of serious games (Marsh, 2011; Miglino et al., 2008, 2007) that could be used to disseminate key concepts also to the general public, within Science Museums and Festivals, and to students of the primary and secondary schools.

### Conclusion

FARSa is an easy to use open-source tool that can enable students and researchers with a limited technical expertise to start building and experimenting with embodied cognitive science models and enables experienced researchers to use a powerful tool that can be easily configured and extended. Moreover, it provides an unique set of integrated tools that strongly facilitate the design of neuro-robotics and adaptive robots. The current version of the tool has been extensively used and tested to carry on frontier research in adaptive robotics (Massera et al., 2007; Gigliotta and Nolfi, 2007; Tuci et al., 2009; Massera et al., 2010; Tuci et al., 2011; Savastano and Nolfi, 2012; Leugger and Nolfi, 2012) in our lab. We hope the public and well-documented version of the tool that we just released will attract a wide interest

and will permit the establishment of a wide community of users and developers.

### Acknowledgments

This research has been supported by CNR, in part under the European Science Foundation project Hierarchical Heterogeneous SWARM (H2Swarm). The authors thank Piero Savastano, Fabrizio Papi, and Tobias Leugger who contributed to the development of the tool.

### References

- Bonani, M., Longchamp, V., Magnenat, S., Re?ornaz, P., Burnier, D., Roulet, G., Vaussard, F., Bleuler, H., and Mondada, F. (2010). The marXbot, a miniature mobile robot opening new perspectives for the collective-robotic research. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4187–4193.
- Carpin, S., Lewis, M., Wang, J., Balakirsky, S., and Scrapper, C. (2007). USARSim: a robot simulator for research and education. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1400–1405. Ieee.
- Clark, A. (1999). An embodied cognitive science? *Trends in cognitive sciences*, 3(9):345–351.
- Gerkey, B. P., Vaughan, R. T., Stø y, K., Howard, A., Sukhatme, G. S., and Mataric, M. J. (2001). Most valuable player: A robot device server for distributed control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, number Iros, pages 1226–1231.
- Gigliotta, O. and Nolfi, S. (2007). Formation of spatial representations in evolving autonomous robots. *2007 IEEE Symposium on Artificial Life*, pages 171–178.
- Jerez, J. and Suero, A. (2004). Newton Game Dynamics. <http://www.newtondynamics.com>.
- Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2149–2154. Ieee.
- Leugger, T. and Nolfi, S. (2012). Action development and integration in an humanoid icub robot. In *Biomimetic and Biohybrid Systems*, pages 369–370. Springer.
- Marsh, T. (2011). Serious games continuum: Between games for purpose and experiential environments for purpose. *Entertainment Computing*, 2(2):61 – 68.
- Massera, G., Cangelosi, A., and Nolfi, S. (2007). Evolution of prehension ability in an anthropomorphic neurobotic arm. *Frontiers in neurobotics*, 1:1–9.
- Massera, G., Tuci, E., Ferrauto, T., and Nolfi, S. (2010). The Facilitatory Role of Linguistic Instructions on Developing Manipulation Skills. *IEEE Computational Intelligence Magazine*, 5(3):33–42.
- Metta, G., Fitzpatrick, P., and Natale, L. (2006). Yarp: yet another robot platform. *International Journal on Advanced Robotics Systems*, 3(1):43–48.
- Michel, O. (2004). Webots: professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42.
- Miglino, O., Gigliotta, O., Cardaci, M., and Ponticorvo, M. (2007). Artificial organisms as tools for the development of psychological theory: Tolman’s lesson. *Cognitive Processing*, 8:261–277.
- Miglino, O., Gigliotta, O., Ponticorvo, M., and Stefano, N. (2008). Breedbot: an evolutionary robotics application in digital content. *The Electronic Library*, 26(3):363–373.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotcz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65.
- Mondada, F., Franzi, E., and Ienne, P. (1994). Mobile robot miniaturisation: A tool for investigation in control algorithms. In *Experimental Robotics III*, pages 501–513. Springer.
- Nolfi, S. (2000). Evorobot 1.1 User Manual. Technical report, Institute of Psychology, CNR.
- Nolfi, S. and Gigliotta, O. (2010). Evolution of Communication and Language in Embodied Agents. pages 297–301.
- Pfeifer, R. and Bongard, J. (2007). *How the body shapes the way we think: a new view of intelligence*. Bradford Books.
- Pinciroli, C., Trianni, V., O’Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Caro, G., Ducatelle, F., Birattari, M., Gambardella, L. M., and Dorigo, M. (2012). ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295.

- Sandini, G., Metta, G., and Vernon, D. (2004). Robotcub: An open framework for research in embodied cognition. In *Proceedings of IEEE/RAS International Conference on Humanoid Robots*, pages 13–32.
- Savastano, P. and Nolfi, S. (2012). Incremental learning in a 14 dof simulated icub robot: Modeling infant reach/grasp development. In *Biomimetic and Biohybrid Systems*, pages 250–261. Springer.
- Shapiro, L. (2007). The Embodied Cognition Research Programme. *Philosophy Compass*, 2(2):338–346.
- Smith, R. (2004). Open Dynamics Engine. <http://www.ode.org>.
- Tikhanoff, V., Cangelosi, A., and Fitzpatrick, P. (2008). An open-source simulator for cognitive robotics research: the prototype of the iCub humanoid robot simulator. In *Proceedings of IEEE Workshop on Performance Metrics for Intelligent Systems Workshop (PerMIS'08)*.
- Tuci, E., Ferrauto, T., Zeschel, A., Massera, G., and Nolfi, S. (2011). An Experiment on Behavior Generalization and the Emergence of Linguistic Compositionality in Evolving Robots.
- Tuci, E., Massera, G., and Nolfi, S. (2009). Active categorical perception in an evolved anthropomorphic robotic arm.

# Insect Type MEMS Micro Robot Controlled by CMOS IC of Hardware Neural Networks

M. Takato<sup>1</sup>, S. Yamasaki<sup>1</sup>, S. Takahama<sup>1</sup>, J. Tanida<sup>1</sup>, K. Saito<sup>2</sup> and F. Uchikoba<sup>2</sup>

<sup>1</sup> Precision Machinery Engineering, Graduate School of Science and Technology, Nihon University

<sup>2</sup>Dept. of Precision Machinery Engineering, Collage of Science and Technology, Nihon University  
takato@eme.cst.nihon-u.ac.jp, kensaito@eme.cst.nihon-u.ac.jp, uchikoba@eme.cst.nihon-u.ac.jp

## Abstract

This paper describes insect type micro robots controlled by a CMOS IC of hardware neural networks. The micro robot is fabricated by the micro electro mechanical systems (MEMS) technology using a silicon wafer, and the actuator is composed of artificial muscle wires on the basis of shape memory alloy. Insect-like walking is achieved by link mechanisms that transform the actuator's rotational motion to locomotive motion. The CMOS IC generates the driving waveform of the micro robot and realizes insect-like walking. The hardware neural networks are built as cell body models and inhibitory synaptic models. The output signal ports of the hardware neural networks are connected to the artificial-muscle-wire-driving circuit. This robot system does not require specialized software programs and A/D converters. The developed neural networks are composed of self-functioning, interconnected plural unit neurons. For proper driving, each neuron in the developed neural network control must be synchronized, as occurs in the neural networks of living organisms. In this study, the motion of the MEMS micro robot is controlled by non-synchronization and anti-phase synchronization driving waveforms. When the non-synchronization driving waveform is input, the micro robot ceases walking motion, but resumes walking upon receipt of the anti-phase synchronization driving waveform. The sideways, endways, and height dimensions of the fabricated micro robot are 4.0 mm, 2.7 mm and 2.5 mm, respectively. The obtained locomotion speed is 26.4 mm/min and the step width is 0.88 mm.

## Introduction

Insect-like micro robots have been increasingly applied in medicine and other fields that require precise manipulation (Shibata et al., 1997; Takeda, 2001; Habib et al., 2007, 2011; Baisch et al., 2010; Yan et al., 2007). However, to date, very few insect robots are operational at the level of living organisms. Developmental obstacles include the miniaturization of the mechanism, small energy requirements for ensuring long lifetime, and realizing the flexibility of living organisms.

In conventional robot mechanisms, the dominant actuator is an electromagnetic motor, and robotic components are manufactured by mechanical machining. However, very small robots cannot be fabricated by conventional technologies. To overcome this limitation, researchers have developed the micro electro mechanical systems (MEMS) technology on the

basis of the IC production process (Donald et al., 2006; Edqvist et al., 2009). The miniature actuators reported to date include electrostatic actuators (Tang et al., 1989; Sniegowski et al., 1996), electromagnetic actuators (Asada et al., 1994), piezoelectric actuators (Suzuki et al., 1999), and shape memory alloy (SMA) actuators (Surbled et al., 2001). However, the movement of micro robots built with these actuators is impeded on uneven surfaces by small gaps or dust particles. Therefore, micro robots that can walk on an uneven surface are highly sought.

Conventional robot control is implemented by digital systems based on microprocessors and software programs. While the pre-programmed digital system exerts adequate control in the specified environment, it may not respond appropriately to unpredictable events. As a means of realizing flexible control, artificial neural networks have attracted considerable attention (Nakada et al., 2003; Delcpyn, 1980). For example, an organism such as an insect realizes walking motion by combining simple neural networks. Moreover, living organisms are flexible and respond sensitively to accidental events. For these reasons, artificial neural network control is a preferable choice in micro robot design.

Neural networks have mostly been investigated by software approaches based on mathematical calculations (Tsumoto et al., 2003, 2006; Tsuji et al., 2007; Hodgkin et al., 1952; FitzHugh, 1961). However, using this approach, even simple neural networks consume vast physical and temporal computer resources. Therefore, several researchers have implemented neural networks in hardware (Endo et al., 1978; Kitajima et al., 2001; Yamauchi et al., 1999, 2003; Nagumo et al., 1962; Maeda, 2008). Hardware networks can process nonlinear operations continuously at high speed. Moreover, because the circuit can be embodied in an integrated circuit (IC) (Lewis et al., 2000), extreme size reduction is expected even for large circuits.

The previous micro robot system reported by the authors (Suematsu et al., 2009; Saito et al., 2010; Okazaki et al., 2011) possessed two legs controlled by the MEMS technology and a plastic body. The actuator was fabricated by SMA. The micro robot was controlled by pulse-type hardware networks (P-HNNs), functioning similarly to the central pattern generator of living organisms. The P-HNN was composed of discrete components. In the next step of the project, the six-legged structure and walking motion of the insect were developed. The rotational actuator used SMA-based artificial



muscle wires, and walking motion was realized by link mechanisms. Motion was controlled by a hardware neural network circuit connected to the actuator. Moreover, the micro robot dimensions were reduced to 4.0 mm, 2.7 mm, 2.5mm in the sideways, endways and height directions, respectively (Uchikoba et al., 2012). In this study, the robot was again controlled by discrete P-HNN circuits.

Recently, we built hardware neural networks into CMOS IC. The IC was connected to the insect-type MEMS micro robot, and its performance was evaluated. In this paper, we explain the mechanism and system of the micro robot and P-HNN IC, and discuss the control characteristics of the P-HNN IC. The driving waveform is provided by synchronization of several hardware neural networks.

## System of Micro Robot

### CMOS IC Controlled MEMS Micro Robot

In future applications, the developed micro robot will observe and possibly lead swarms of similar micro robots. Therefore, the robot should resemble a live insect as much as possible. For this purpose, we consider hexapod walking motion as a primary objective. Figure 1 shows the mechanism of the designed micro robot.

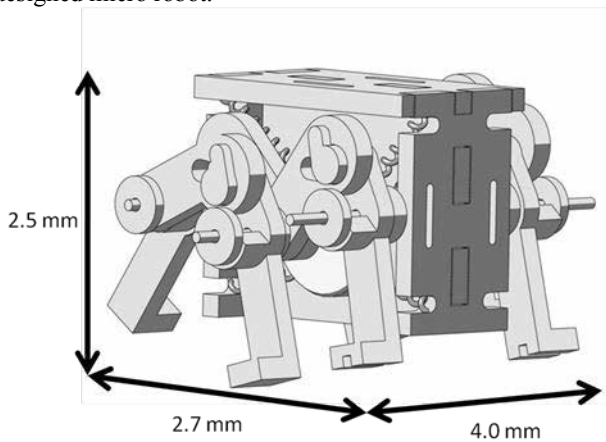


Figure 1: Mechanism of the Designed Micro Robot

The robot is fitted with six legs on either side. Three of the legs are connected by the link mechanism, while the central leg is held on the rotational actuator built into the body parts. The rotor of the actuator is suspended by artificial muscle wires extending in four directions. The artificial muscle wires, composed of SMA, shrink when heated above the transition temperature of the alloy, and revert to their original length by cooling below the transition temperature. The temperature is raised by flowing an electrical current directly into the wire. The wire is coiled inside the actuator to enable a larger displacement than is possible with a linear wire. The specifications of the artificial muscle wire are shown in Table 1 (<http://www.toki.co.jp>).

Coil diameter (mm)	0.2
Wire diameter (mm)	0.05
Drive current (mA)	50–120
Resistance ( $\Omega\text{m}^{-1}$ )	3600
Force (gf)	3–5
Displacement (%)	50

Table 1: Specifications of Artificial Muscle Wire

Rotational motion is realized by shrinking the artificial muscle wires in rotational order. A schematic of the actuator's rotational motion is shown in Figure 2. As wire A shrinks under heating, the rotor is pulled toward the A side. In the next step, wire A is extended by cooling, and wire B is shrunk, dragging the rotor toward the B side. The heating and cooling processes are repeated for wires C and D. These activities induce a clockwise rotation of the rotor.

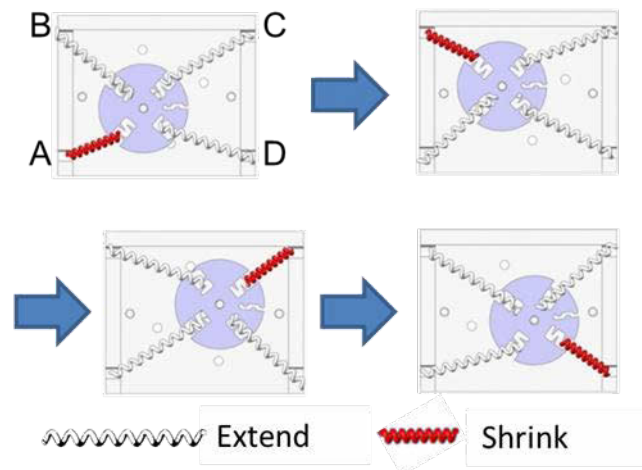


Figure 2: Schematic of Clockwise Rotational Motion of the Actuator

The driving waveform of the neural networks for generating hexapod walking motion is shown in Figure 3. The walking motion is generated by four pulses. The input voltage and pulse width are 3 V and 0.5 s, respectively.

The central leg is connected to the rotor shaft. The front and rear legs float, while the central leg touches the ground surface. Governed by the rotational motion of the actuator, the middle leg kicks the ground, propelling the micro robot body forward. As the middle leg rises, the front and rear legs touch the ground surface. The outer and central legs are out of phase by 180°. During these motions, the touched points always form a triangle, and stable walking motion, mimicking that of an insect, is realized. Moreover, backward locomotion is generated by reversing the actuator motion. The forward and backward motions are controlled by the electrical current pulses shown in Figure 3 (a) and 3 (b), respectively.

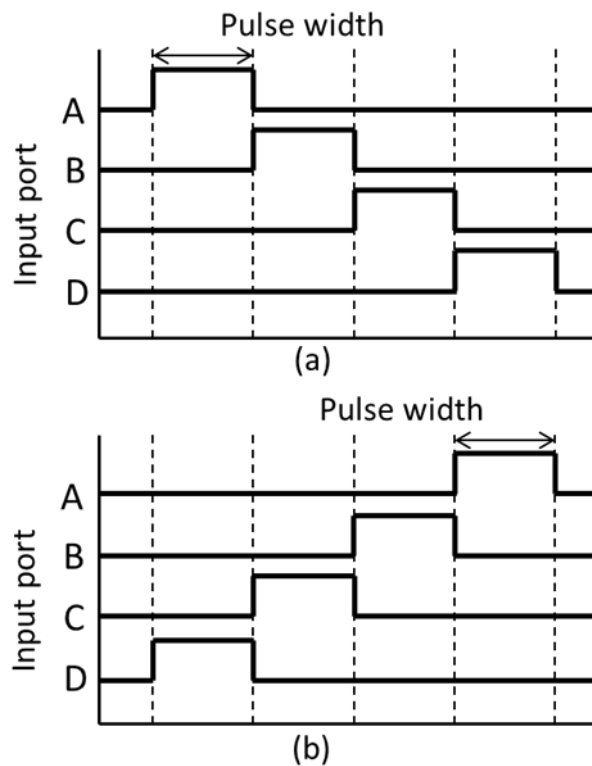


Figure 3: Schematic of Driving Waveform: (a) Clockwise Pulse (b) Reverse Rotation Pulse

### Structure of Micro Robot

The fabricated micro robot comprises the body parts, the rotational actuator, and the legs. The body, legs, and actuator rotor are made from single crystal silicon wafers fabricated by the MEMS technology. The four frames of the body structure are assembled by jutting the parts into the grooves. For this purpose, four pieces of artificial muscle wires are connected to the rotor section of the rotational actuator, which is modified by zigzag slits. An electrical ground line is led from the rotor. The other end of the artificial muscle wire is connected to the frame. Figure 4 shows a schematic of the micro robot body and the actuator part, while the shape and dimensions of the rotor part are shown in Figure 5.

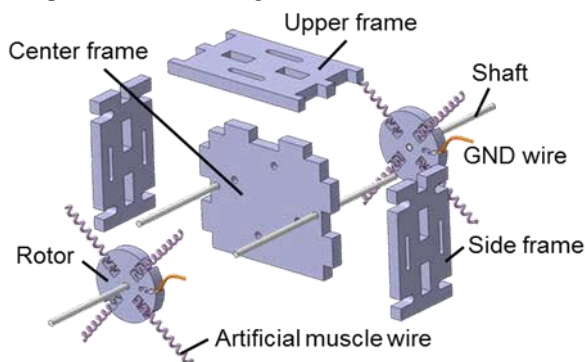


Figure 4: Schematic of Micro Robot Body and Actuator

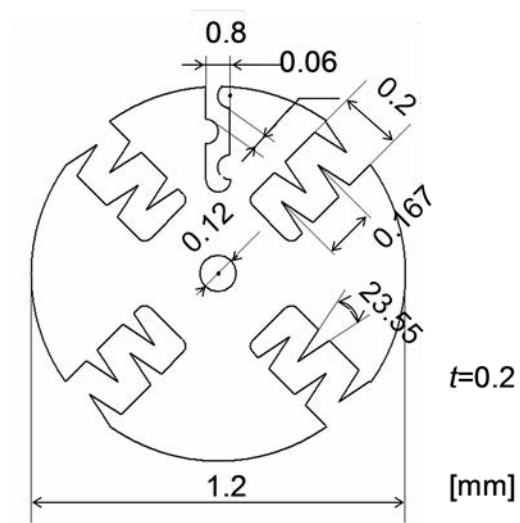


Figure 5: Shape and Dimensions of Rotor Part

The three leg parts are connected by the link mechanism. The front and rear legs move in the counter-direction relative to the center leg. To ensure that the robot moves parallel to the ground, the central leg is made shorter than the other legs. Figure 6 shows the link mechanism of the legs. The central legs are connected to the center of the rotor by a tungsten carbide shaft, while the outer legs are held by shafts attached to the body frame.

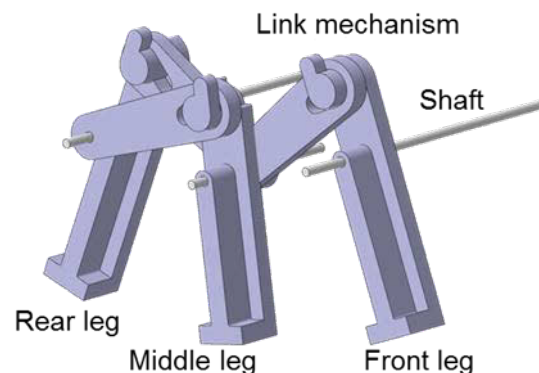


Figure 6: Link Mechanism of Legs

The silicon part was fabricated by MEMS photolithography. After washing the silicon wafer, aluminum was deposited by physical vapor deposition, followed by coating with a photoresist. The aluminum film was approximately  $0.1 \mu\text{m}$  thick. The designed pattern was exposed to the resist film, and developed by soaking in the developer. The aluminum film on the specimen was then etched chemically, leaving an imprint of the designed pattern. The washed and dried specimen was dry-etched by high-aspect-ratio induced coupled plasma etching combined with a Bosch process (Bhardwaji et al., 1995). The rotor part was obtained after removing the aluminum film and washing. The other robot parts were obtained by repeating this process on both surfaces of the wafer. Hand assembly of the fabricated parts yielded the micro robot.

### Pulse-Type Hardware Neural Networks

The P-HNN was built from the cell body and synaptic model circuits. These circuits reproduce the functions of biological neurons.

The cell body model circuit is configured as a voltage control negative resistance circuit, an equivalent inductance circuit, membrane capacitor  $C_M$ , and leak resistor  $M_{C4}$ . The cell body model circuit is characterized by a firing threshold, refractory period, and a continuous pulse waveform, similar to the characteristics of biological neurons. Figure 7 is a diagram of the cell body model circuit, powered by source  $V_A$ .

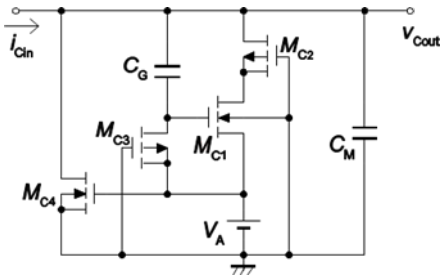


Figure 7: Diagram of the Cell Body Model Circuit

The synaptic model circuit comprises the excitatory and inhibitory synaptic model circuits, which differ only in the direction of their current. The synaptic model circuit has spatio-temporal summation characteristics mimicking those of biological synapses. That is, the synaptic model circuit sums the output pulses of the cell body model circuit. A diagram of the synaptic model circuit is provided in Figure 8. This circuit is powered by source  $V_{DD}$ . The synaptic weight (connection strength between the cell body models) is adjustable by changing the ration of the channel length  $L$  and the channel width  $W$  in the gate of  $M_{IS1}$  in Figure 8 (b). For example,  $i_{ISout}$  is increased by increasing the  $W/L$  of  $M_{IS1}$ .

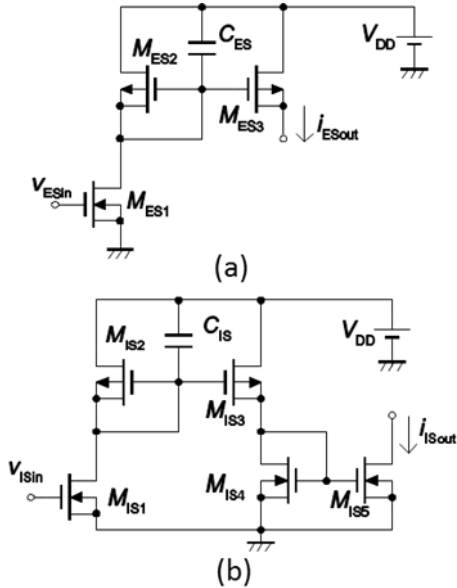


Figure 8: Diagram of the Synaptic Model Circuit: (a) Excitatory Synaptic Model Circuit (b) Inhibitory Synaptic Model Circuit

The P-HNNs are synchronized by the excitatory and inhibitory synaptic model circuits. Excitatory and inhibitory mutual coupling generates in-phase and anti-phase synchronization, respectively. During walking, we consider that the neural networks of the micro robot might consist solely of inhibitory synaptic models and anti-phase synchronization. If the excitatory synaptic model can be removed without loss of functionality, the number of elements in the model can be reduced. The P-HNN connections are schematized in Figure 9.

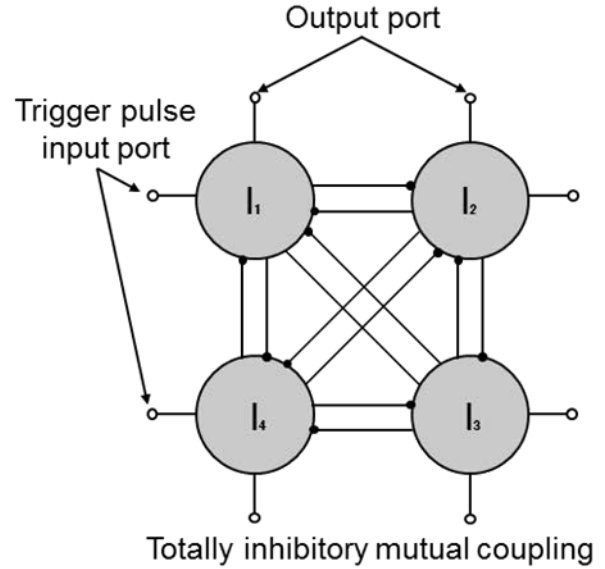


Figure 9: Connection Diagram of P-HNN

The four cell body models are mutually coupled by 12 inhibitory synaptic models. Four output ports, extracted from the P-HNN, are connected to the actuator of the MEMS micro robot. In addition, the trigger pulse input ports are extracted to P-HNNs. By accepting different input timing of the external trigger pulse, the P-HNN can alter the sequence of the output waveform.

### CMOS IC

Figure 10 displays the layout pattern of the P-HNN. The process line is a double-metal single-poly CMOS 0.35  $\mu\text{m}$  rule. The area of the IC chip is 1.93 mm square. Because capacitors  $C_G$  and  $C_M$  are too large to position on the CMOS IC chip, they are externally connected to the chip; this arrangement also enables straightforward dynamic frequency adjustment. Figure 11 shows a diagram of the artificial muscle wire driving circuit that generates the walking motion of the MEMS micro robot.



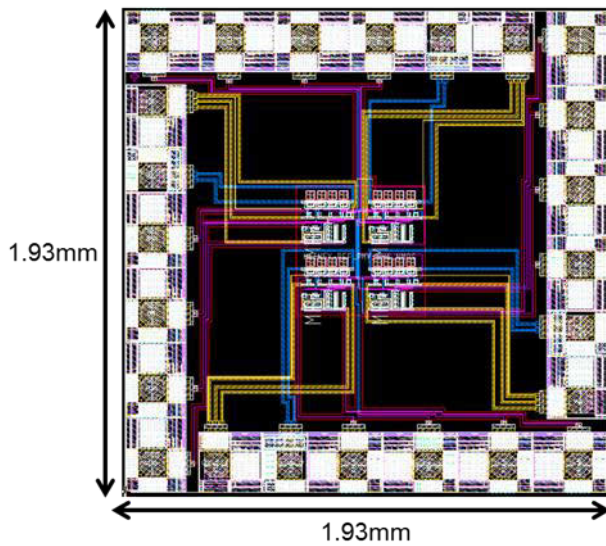


Figure 10: Layout Pattern of P-HNN

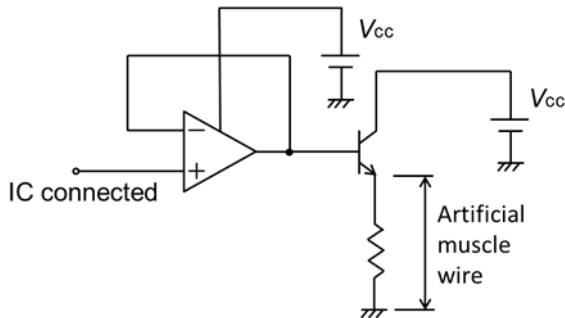


Figure 11: Artificial Muscle Wire Driving Circuit

## Results and Discussion

A photograph of the fabricated micro robot is shown in Figure 12. The sideways, endways, and height dimensions of the micro robot are 4.0 mm, 2.7 mm, and 2.5 mm, respectively. The dimensional error of the actuator component was measured by an optical con-focal microscope, and was found to be always within  $\pm 3 \mu\text{m}$ . Moreover, the leg and link parts were connected with adequate clearance fit (measured as 20–30  $\mu\text{m}$ ).

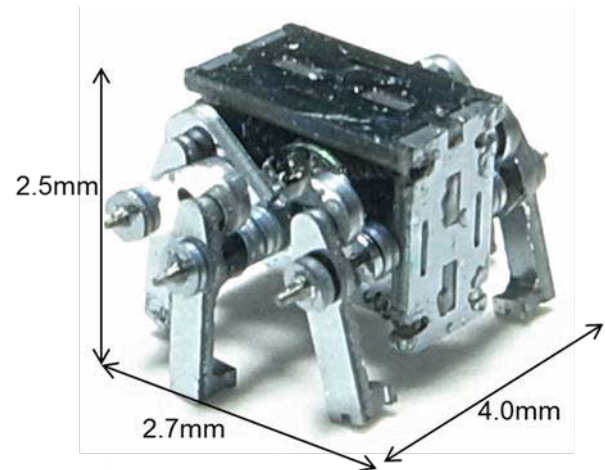


Figure 12: Photograph of the Fabricated MEMS Micro Robot

Figure 13 compares the discrete circuit board of the artificial neural networks with the packaged IC. This figure shows that the packaged IC was miniaturized to 3.5% of the discrete circuit area, while the bare die realized an area reduction to 0.05%. Thus, considerable size reduction may be achieved by the IC construction.

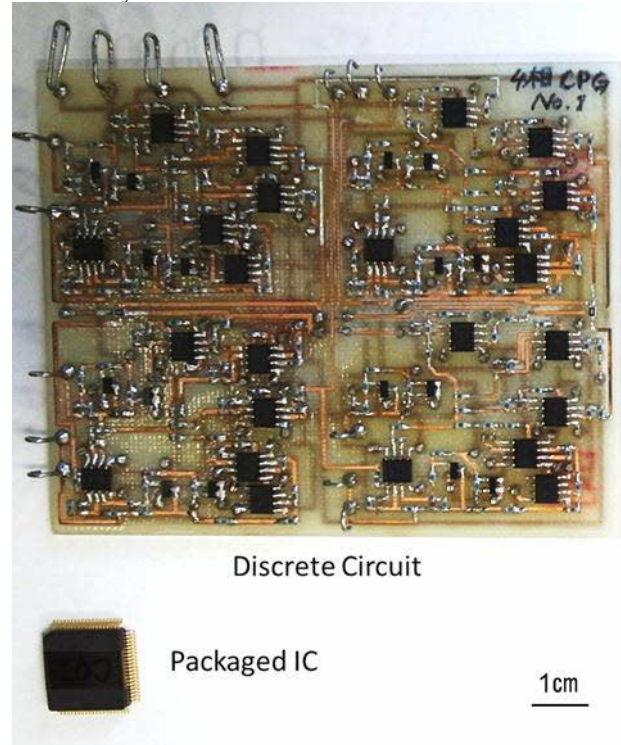


Figure 13: Size Comparison between the Discrete Circuit and the Packaged IC Discrete Circuit: 10 cm  $\times$  8 cm, Packaged IC: 14 mm  $\times$  20 mm, Bare Die IC: 1.93 mm  $\times$  1.93 mm

Figure 14 shows the output waveform of the IC chip when  $V_{DD} = 5 \text{ V}$ . At this voltage level, the inhibitory synaptic circuit is turned on. In other words, the inhibitory synaptic model circuit is connected, generating an output waveform that inhibits the other cell body model circuits. In response, the cell



body model circuit comprising  $I_1$ ,  $I_2$ ,  $I_3$ , and  $I_4$  outputs an anti-phase synchronization pattern. Since P-HNNs can generate driving waveforms such as those shown in Figure 3, we have demonstrated that the fabricated IC chip generates correct walking motion of the silicon micro robot.

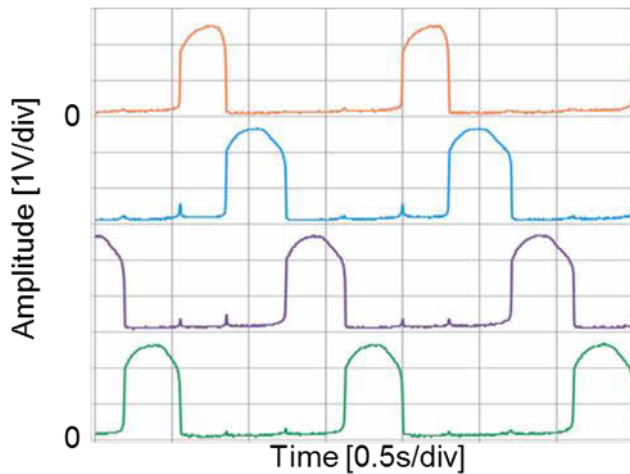


Figure 14: Output Waveform of the IC Chip

Figure 15 illustrates the walking motion of the MEMS micro robot. The driving waveform that moves the robot is output by the P-HNN. The rotary actuator is then activated and the link mechanism is converted to a walking motion. The locomotion speed is 26.4 mm/min with a step width of 0.88 mm.

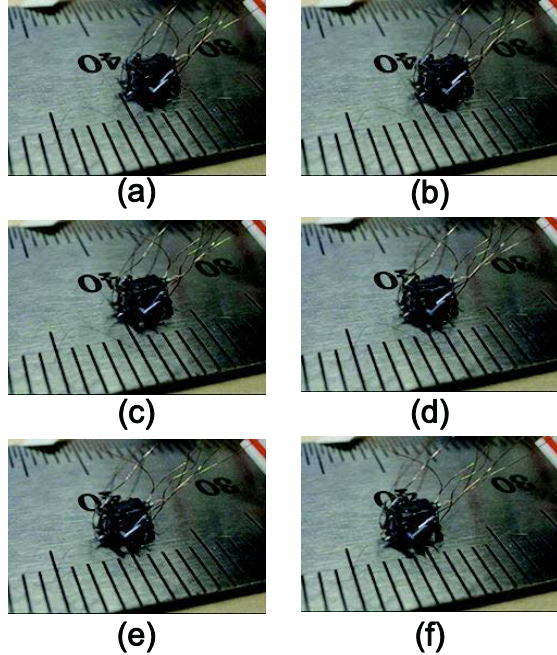


Figure 15: Walking Motion of MEMS Micro Robot

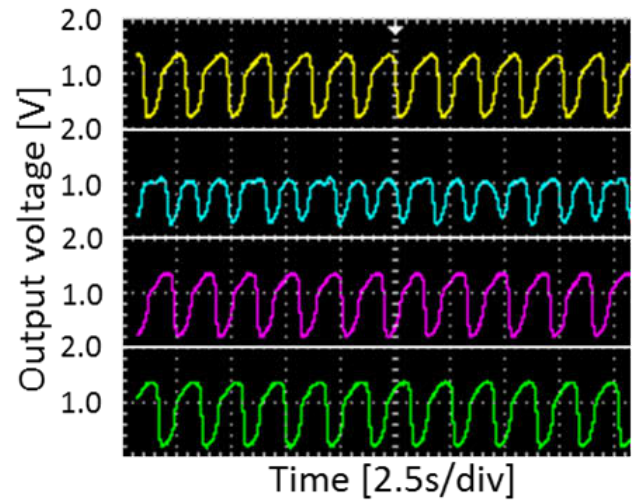


Figure 16: Output Waveform of IC (Non-Synchronization Mode)

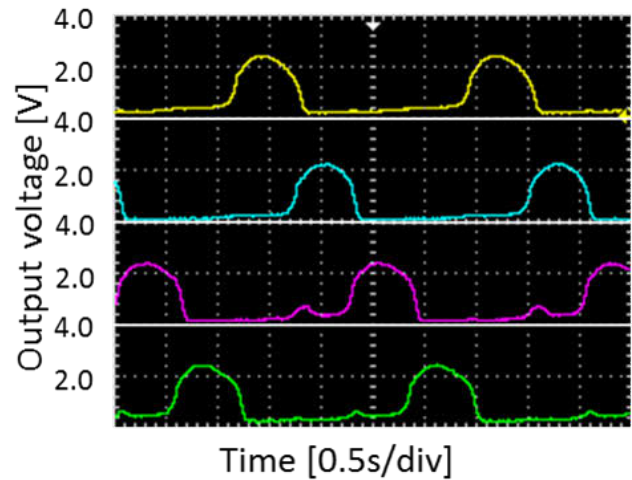


Figure 17: Output Waveform of IC (Anti-Phase Synchronization Mode)

The motion of the MEMS micro robot was examined under non-synchronization and anti-phase synchronization P-HNN driving waveforms, displayed in Figs. 16 and 17, respectively. In Figure 16, four waveforms are randomly generated, while in Figure 17, they are alternately generated. In the former case, the actuator of the micro robot did not perform rotational motion, hence the robot did not achieve walking function. By contrast, when the anti-phase synchronization driving waveform was connected to the micro robot, walking motion was initiated (Figure 18). To easily visualize the leg movements, these motional comparisons were conducted on a larger micro robot than the fabricated one. The width, length, and height of this micro robot were 8.1 mm, 8.9 mm, and 9.0 mm, respectively. Each figure was observed at 0.5 s intervals. The locomotion speed of this micro robot was 96.0 mm/min with a step width of 4.0 mm. In this test, the MEMS micro robot was controlled by the CMOS IC of hardware neural

networks, whose synchronization phenomena mimicked those of brain networks.

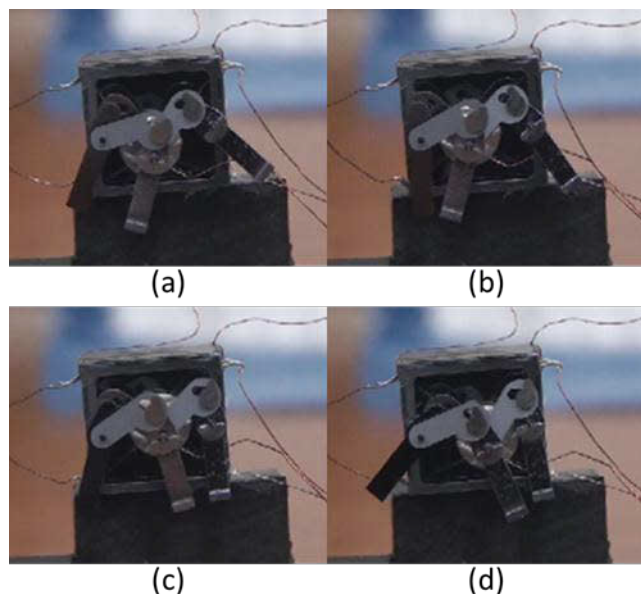


Figure 18: Motion Response of MEMS Micro Robot to an Anti-Phase Synchronization Driving Waveform

## Conclusion

This paper proposes an insect-type MEMS micro robot controlled by CMOS IC of hardware neural networks. The micro robot contains a rotational actuator using SMA-based artificial muscle wires and possesses the hexapod structure of an insect. The dimensions of the micro robot are 4.0 mm (sideways), 2.7 mm (endways), and 2.5 mm (height).

The micro robot walks when supplied with a synchronized output driving waveform generated by the IC chip of pulse-type hardware neural networks. Thus, as in living organisms, walking motion is realized by the synchronization of artificial neural networks. Moreover, the movement of the micro robot can be controlled by pulse-type hardware neural networks. The locomotion speed is 26.4 mm/min and the step width is 0.88 mm.

In the future, other motions exhibited by living organisms will be incorporated into the insect-type MEMS micro robot, by integrating functional elements such as sensors onto a single chip. When the sensor has been integrated, the walking motion systems will form a closed loop, enabling more lifelike behavior of the micro robot. Such micro robots are not only useful in medical fields but will also assist biologists in understanding the movements of living organisms. Moreover, since the all-in-one package installed in the micro robot requires a built-in battery, the micro robot may be used to investigate electromagnetic induction-type wireless power transfer or solar cells.

## Acknowledgments

Specimen fabrication was supported by Research Center for Micro Functional Devices of Nihon University. This study was supported by the CST research project of Nihon University, and by JSPS KAKENHI (25420226, 23760243).

## References

- Doi, S. Kumagai, S. editors (1962). *Nonlinear dynamics of small-scale biophysical neural networks*. Foundations of Integrative Neuroscience, Mary Ann Liebert, page 261-301.
- Shibata, T. Aoki, Y. Otsuka, M. Idogaki, T. Hattori, T. (1997). Microwave Energy Transmission System for Microrobot. IEICE transactions on electronics, E80-C(2):303-308.
- Habib, M. K. (2011). Biomimetics: Innovations and Robotics. International Journal of Mechatronics and Manufacturing Systems, 4:113-134.
- Donald, B. R. Levey, C. G. McGray, C. D. Paprotny, I. Rus, D. (2006). An Untethered, Electrostatic, Globally Controllable MEMS Micro-Robot. Journal of Microelectromechanical Systems, 15:1-15.
- Edqvist, E. Snis, N. Mohr, R. C. Scholz, O. Corradi, P. Gao, J. Diéguez, A. Wyrsh, N. Johansson, S. (2009). Evaluation of Building Technology for Mass Producing Millimeter-Sized Robots using Flexible Printed Circuit Boards. Journal of Micromechanics and Microengineering, 19:1-11.
- Asada, N. Matsuki, H. Minami, K. Esashi, M. (1994). Silicone Micromachined Two-Dimensional Galvano Optical Scanner. IEEE Transactions on Magnetics, 30:4647-4649.
- Surbled, P. Clerc, C. Le Prouff, B. Ataka, M. Fujita, H. (2001). Effect of the Composition and Thermal Annealing on the Transformation Temperature Sputtered TiNi Shape Memory Alloy Thin Films. Thin Solid Films, 401:52-59.
- Nakada, K. Asai, T. Amemiya, Y. (2003). An Analog CMOS Central Pattern Generator for Interlimb Coordination in Quadruped Locomotion. IEEE Transaction on Neural Networks, 14:1356-1365.
- Delcomyn, F. (1980). Neural Basis of Rhythmic Behavior in Animals. Science, 210:492-498.
- Tsumoto, K. Yoshinaga, T. Aihara, K. Kawakami, H. (2003). Bifurcations in Synaptically Coupled Hodgkin-Huxley Neurons with a Periodic Input. International Journal of Bifurcation and Chaos, 13:653-666.
- Tsuji, S. Ueta, T. Kawakami, H. Aihara, K. (2007). Bifurcation Analysis of Current Coupled BVP Oscillators. International Journal of Bifurcation and Chaos, 17:837-850.
- Tsumoto, K. Yoshinaga, T. Iida, H. Kawakami, H. Aihara, K. (2006). Bifurcations in a Mathematical Model for Circadian Oscillations of Clock Genes. Journal of Theoretical Biology, 239:101-122.
- Hodgkin, A. L. Huxley, A. F. (1952). A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve. The Journal of Physiology, 117:500-544.
- FitzHugh, R. (1961). Impulses and Physiological States in Theoretical Models of Nerve Membrane. Biophysical Journal, 1:445-466.
- Endo, T. Mori, S. (1978). Mode Analysis of a Ring of a Large Number of Mutually Coupled van der Pol Oscillators. IEEE Transactions on Circuits Systems, 25:7-18.
- Kitajima, H. Yoshinaga, T. Aihara, K. Kawakami, H. (2001). Burst Firing and Bifurcation in Chaotic Neural Networks with Ring Structure. International Journal of Bifurcation and Chaos, 11:1631-1643.
- Yamauchi, M. Wada, M. Nishino, Y. Ushida, A. (1999). Wave Propagation Phenomena of Phase States in Oscillators Coupled by Inductors as a Ladder. IEICE Trans. Fundamentals, E82-A:2592-2598.
- Yamauchi, M. Okuda, M. Nishino, Y. Ushida, A. (2003). Analysis of Phase-Inversion Waves in Coupled Oscillators Synchronizing at In- and Anti-Phase. IEICE Trans. Fundamentals, E86-A:1799-1806.
- Okazaki, K. Ogiwara, T. Yang, D. Sakata, K. Saito, K. Sekine, Y. Uchikoba, F. (2011). Development of Pulse Control Type MEMS Micro Robot with Hardware Neural Network. Artificial Life and Robotics, 16:229-233.
- Uchikoba, F. Takato, M. Saito, K. (2012). Hardware Neural Networks Controlled MEMS Rotational Actuators and Application to Micro

- Robot. Journal of Mechanics Engineering and Automation, 2:499–506.
- Sniegowski, J. J. Garcia, E. J. (1996). Surface-micromachined Gear Trains Driven by an On-Chip Electrostatic Microengine. *IEEE Electron Device Letters*, 17:366–368.
- Takeda, M. (2001). Applications of MEMS to Industrial Inspection. In *Proceedings of MEMS 2001*, pages 182–191.
- Habib, M. K. Watanabe, K. Izumi, K. (2007). Biomimetics robots: From Bio-inspiration to Implementation. In *Proceedings of 33rd Annual Conference of the IEEE Industrial Electronics Society*, pages 143–148.
- Baisch, A. T. Sreetharan, P. S. Wood, R. J. (2010). Biologically-Inspired Locomotion of a 2g Hexapod Robot. In *Proceedings of IEEE IROS 2010*, pages 5360–5365.
- Yan, G. Ye, D. Zan, P. Wang, K. Ma, G. (2007). Micro-Robot for Endoscope Based on Wireless Power Transfer. In *Proceedings of 33rd Annual Conference of the 2007 IEEE International Conference on Mechatronics and Automation*, pages 3577–3581.
- Tang, W. C. Nguyen, T.-C. H. Howe, R. T. (1989). Laterally Driven Polysilicon Resonant Microstructure. In *Proceedings of IEEE Micro Electro Mechanical Systems. An Investigation of Micro Structures, Sensors, Actuators, Machines and Robots*, pages 53–59.
- Suzuki, Y. Tani, K. Sakuhara, T. (1999). Development of a New Type Piezo Electric Micromotor. In *Proceedings of Transducers '99*, pages 1748–1751.
- Nagumo, J. Arimoto, S. Yoshizawa, S. (1962). An Active Pulse Transmission Line Simulating Nerve Axon. In *Proceedings of IRE*, pages 2061–2072.
- Maeda, Y. (2008). A Hardware Neuronal Network Model of a Two-level Central Pattern Generator. Transactions of the Japanese Society for Medical and Biological Engineering, 46:496–504.
- Lewis, A. M. Cummings, E. R. Cohen, H. A. Hartmann, M. (2000). Toward Biomimetic Control Using Custom a VLSI CPG Chips. In *Proceedings of International Conference on Robotics & Automation*, pages 494–500.
- Suematsu, H. Kobayashi, K. Ishii R. Matsuda, A. Skine, Y. Uchikoba, F. (2009). MEMS Type Micro Robot with Artificial Intelligence System. In *Proceedings of International Conference on Electronics Packaging*, pages 975–978.
- Saito, K. Okazaki, K. Kawakami, T. Matsuda, A. Uchikoba, F. Sekine, Y. (2010). Pulse-Type Hardware CPG Model for MEMS Type Micro Robot. In *Proceedings of 2010 IEEE International Analog VLSI Workshop*, pages 219–223.
- Bhardwaj, J. K. Ashraf, H. (1995). Advanced Silicon Etching using High-Density Plasmas. In *Proceedings of SPIE Micromachining and Microfabrication Process Technology*, pages 224–233.

# Open-Ended Evolution of a Circadian Rhythm

Tiago Baptista and Ernesto Costa

CISUC, University of Coimbra, Coimbra, Portugal  
baptista@dei.uc.pt

## Abstract

Most biological systems have some sort of adaptation to our planet's cycle of day and night. This adaptation is a current subject of scientific research, and serves as inspiration to develop a multi-agent simulation to investigate the evolution of complexity in an open-ended evolutionary framework. In a previous work, we created a simulated world where artificial organisms evolve to synchronize with a daily cycle of light and darkness. A multi-agent, artificial life framework was used to implement these simulations. In this paper, we further develop that world, by adding caves to the environment. When in these caves, the agents will perceive a low level of light, as if it were night. This adds an extra layer of complexity to the desired behavior of the agents, as now they need to distinguish "night" from "cave". Using the same agent structure, and the same open-ended evolution framework, we show that the agents evolve to adapt to this new environment. We also show how the agents adapt to the environment with caves, by analyzing their brains.

## Introduction

The study of circadian clocks and similar synchronization phenomena in biological systems is a current subject of scientific research (Rand et al., 2006; Strogatz, 2004). Despite having been extensively studied, these phenomena still have much to be investigated. Our goal, however, is not to learn more about this biological process, but to use it as an inspiration to study the emergence of complex behaviors in an open-ended evolution scenario. To that end, we implement a simulation where the environment has a day and night cycle, and analyze the evolution of the agents' behavior, and their adaptation to this cycle.

In a previous paper, the authors presented some experiments done with such a scenario (Baptista and Costa, 2008), and showed that the agents do develop behaviors adapted to the daily cycle. In an effort to create a more challenging environment, we now further developed that world by adding caves. When an agent enters a cave, the light level will be the same as if it were night time. This will force the agents to evolve behaviors capable of distinguishing the two different low light conditions, adding extra complexity to the requirements for survival.

Some previous work has been done with similar simulation scenarios, either by evolving neural networks (Mirolli and Parisi, 2003), or virtual CPU organisms in Avida (Beckmann et al., 2007). Our scenario can be mostly compared to that of (Mirolli and Parisi, 2003), as they also have an environment with varying light level and caves. However, they use a standard genetic algorithm to evolve the agents, whereas we use an open-ended evolution framework.

Although an established definition of open-ended evolution hasn't yet surfaced, most authors consider that one of the major requirements is the absence of an explicit fitness function. In other words, to have open-ended evolution, a system should be based on Natural Selection rather than Artificial Selection (Channon, 2000).

The simulations described in this paper were implemented using the BitBang framework. One of the purposes of these simulations is to serve as a proof of concept for the model developed for the framework. Implementing a modern autonomous agent model (Russell and Norvig, 2002), this framework has roots in Artificial Life systems and Complexity Science. The simulated world is composed of entities. These can either be inanimate objects which we designate as *things*, or entities that have reasoning capabilities and power to perceive and affect the world—the *agents*. Both have traits that characterize them, such as color, size, or energy—the *features*. The agents communicate with, and change the environment using *perceptions* and *actions*, taking decisions using the *brain*. In this model, there is no definition of a simulation step, as we won't have any type of centralized control. As such, the simulation is asynchronous. The agents will independently perceive, decide, and act. Moreover, there is no evolutionary mechanism included in the definition of the model, since evolution is implemented as an action. That is accomplished by giving the agents the capability of reproduction. Again, there is no central control bound to the process of reproduction. The agents choose when to reproduce and with what other agent to reproduce with. In addition, there is no explicit fitness function. The agents die due to lack of resources, predators, age, or any other mechanism implemented in the world. Thus, in this



model we have open ended evolution. To have a more in-depth view of the conceptual model and architecture of Bit-Bang, refer to (Baptista et al., 2006).

In the next section we will describe the simulation world developed, detailing the agents, things, brain architecture, evolutionary process, and environmental settings. We will then present the experimental results and end with some conclusions.

## The DayNight World

In this section we will set out all the implementation details and architecture of the simulations. As mentioned above, these simulations were implemented using the Bit-Bang framework, and therefore we will present the architecture according to the framework's specifications. We begin by describing the simulation environment, then detail the agents' architecture (features, perceptions, actions, and brain). Next, we will present the two types of things defined in this world, and finally we present the architecture of brain used in these experiments.

### The Environment

Our world is a 3D world where agents and resources are placed (see figure 1). The terrain is a square. This area restricts the placement of agents, caves, and resources, but does not restrict the movement of the agents. The world is infinite, i.e., an agent can move past the boundaries of the populated terrain. At startup, the field is populated with a configured amount of randomly placed food items. These are periodically replenished so that the total food count is maintained. The number of resources available is configurable to be able to fine tune the system so as to allow agents to survive but also provide enough evolutionary pressure.

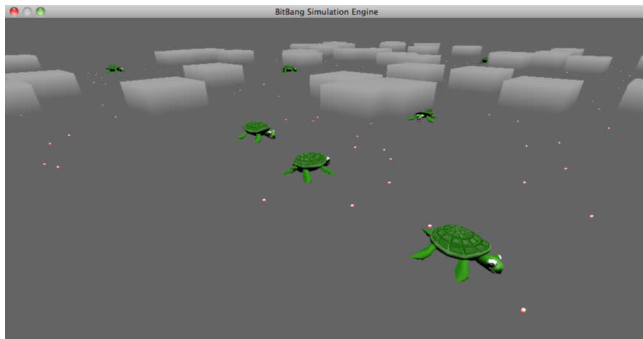


Figure 1: Screenshot of a running simulation. We can see the agents (turtles), the edible resources (small red cubes), and the caves (large grey cubes).

On initialization, the world is populated with randomly placed, and randomly generated agents. At this time, it is highly probable that the agents will not execute the reproduction action, either by not choosing it, or because they

don't have enough energy to reproduce. To keep the population alive, whenever the number of agents in the world falls below a given threshold, new agents are created. If there are live agents in the environment, one will be picked for reproduction, otherwise a new random agent is created. Note that, as stated before, there is no explicit fitness function, so the agent chosen for reproduction will be randomly selected from the population.

To differentiate the day from the night, the environment has a light level that oscillates between a configurable maximum and minimum. For each day the maximum light level is randomly calculated as the overall maximum minus a random value between zero and the delta. The same applies for the day's minimum. For example, if the maximum light level is 100, the minimum light level is 0, and the delta is 10, each day's maximum light level will be a random value between 90 and 100, and each day's minimum light level will be a random value between 0 and 10. Additionally, the light level does not rise or fall abruptly, but rather changes linearly during a specified time interval, simulating dusk and dawn. To better illustrate, in figure 2 this variation of the light level can be observed. The duration of one day, can be configured, and remains constant for the duration of the experiment.

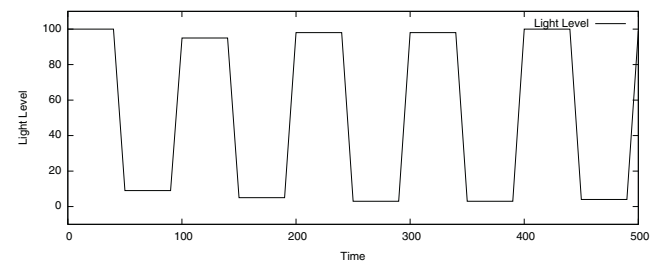


Figure 2: Example of the variation of the light level over the course of five days. In this example, the maximum light level is 100, the minimum is 0, and the delta is 10.

The total simulation time for these experiments was divided into two equal parts. For the first part of the simulation, the agents live in an environment that only has food items. Then, we randomly place in the environment a number of caves, and continue the simulation in this new environment. When an agent enters a cave, the light level it perceives will be the same as if it were night, creating a new challenge for our agents. By the time the caves are created, the agents will have evolved a behavior adapted to the light level, sleeping when the light is low, and being active when its high. However, when an agent enters a cave, it will not be able to distinguish if the low light level means "night" or "cave". If it simply goes to sleep whenever the light level is low, it will never wake up again, as the light level in the cave will never rise. Our agents will now have to adapt their behavior to this new environment.

## The Agents

In this simulation only one type (species) of agent exists, and has the following architecture:

- **Features:** energy, metabolic rate, and birth date.
- **Perceptions:** energy, resource location, reach resource, light level.
- **Actions:** move front, turn left, turn right, sleep, eat, reproduce.
- **Brain:** rule list (see section ).

We will now describe each one of these components.

### Features

**Energy** This feature represents the current energy level of the agent. When this feature reaches zero, the agent dies. The feature is initialized with a predetermined value at agent birth. For these simulations, the agents are initialized with 10 energy units.

**Metabolic Rate** The metabolic rate is the amount of energy the agent consumes per time unit. This rate is initialized to its configured base value, and changes as the agent moves or sleeps. The increase or decrease amounts for move and sleep are configurable.

**Birth Date** This feature is set to the current time at birth and remains constant. It is used to calculate the agent's age. When the agent reaches a given age, it dies. This procedure allows the evolution to continue past the moment when the agents have developed good navigation and eating capabilities, whilst maintaining an asynchronous and open-ended simulation. The maximum age of the agents is configurable.

### Perceptions

**Energy** This is a self-referencing perception on the agent's current energy level. This perception is tied to the corresponding feature. This is a numerical perception, and the range of values can be configured.

**Resource Location** This is the agent's main perception of vision, representing the position of the nearest resource, relative to the agent's position and orientation. The agent's vision is implemented as a 3D cone in front of the agent. The vision cone is configured with a given range and angle, representing its height and aperture. This is a numerical perception with possible values 0, 1, 2, and 3. The value 0 means no resource is visible. The value 1 means there is a resource to the left. The value 2 means there is a resource directly in front of the agent. The value 3 means there is a resource to

the right. This perception is influenced by the light level of the environment. As the light level drops, so does the range of vision for the agent, using the following equation:

$$V(t) = V_0 \frac{L(t)}{L_0}, \quad (1)$$

where  $V(t)$  is the vision range at time  $t$ ,  $V_0$  is the configured vision range of the agents,  $L(t)$  is the light level at time  $t$ , and  $L_0$  is the configured maximum light level.

**Reach Resource** This is a boolean perception that evaluates to true whenever the agent has a resource within its reach. The distance the agent can reach is configurable.

**Light Level** This perception gives the agent the power of sensing the brightness of the environment. This can also be considered a perception of vision. The value of the perception is numeric and, at each time, is evaluated to the environment's current light level if the agent is outside. When in a cave, the perception evaluates to the minimum light level whether it is night or day.

### Actions

**Movement** We define three actions for movement. One to walk forward, one to turn left, and one to turn right. These actions have a tie to the metabolic rate feature in such a way that whenever the agent is moving, the metabolic rate increases.

**Eat** This action enables the agent to eat a resource within its range. If no resource is in range when the action is executed, nothing happens. This action will add a configured amount of energy to the agent's energy feature.

**Sleep** The agent can use this action to sleep. In this simulation, when an agent is sleeping, it will stand still and its metabolic rate will decrease, falling below the base metabolic rate and thus allowing the agent to conserve energy. As for the rest of the actions, it gets executed whenever the agent chooses to do so.

**Reproduce** This action allows the agent to reproduce itself. The reproduction implemented is asexual. When the action is executed, a new agent is created and placed in the world. The new agent will be given a brain that is a mutated version of its parent's brain. Note that, as each mutation operator has a given probability of being applied, the child's brain can be a perfect clone of its parent's brain. The action will also transfer energy from the parent to the offspring. The amount of energy consumed in the action is the sum of the initial energy for the new agent and a configurable fixed cost. It's important to have a cost of reproduction higher than

the initial energy of an agent, so as to provide evolutionary pressure.

**Brain** The agents' brain used in these experiments is a rule list. The architecture of this system is explained in section . On initial creation of an agent, the brain is randomly initialized. This initialization conforms to some configurable parameters: the maximum number of rules, the minimum number of rules, and the maximum number of conditions per rule. Other configured values are the mutation probabilities used in the reproduction action.

### The Things

Two types of things have been defined for this world: the resources that the agents eat to acquire energy, and the caves. No features are associated with them. A configurable parameter defines the amount of energy each resource provides. Although the caves are also represented as things, they are not visible to the agents, or else, it would be easier for them to distinguish "night" and "cave".

### The RuleList Brain

The Rule List brain is composed of an ordered list of rules. The reasoning process is straightforward. The rules are evaluated in order, and the first one whose conditions are all true, is selected. Each rule is composed of a conjunction of conditions and an action. The structure of a rule is shown in listing 1. Next, to illustrate, in listing 2 we provide an example of a rule.

#### Listing 1 Syntax of a rule in the RuleList brain.

```
<rule> ::= IF <cond-list> THEN <action>
<cond-list> ::= <condition>
<cond-list> ::= <condition> AND
<cond-list>
<condition> ::= <percept> <operator>
<percept>
<condition> ::= <percept> <operator>
```

#### Listing 2 Example of a rule.

```
IF energy < 10 AND reach_resource TRUE
THEN eat
```

The use of this brain architecture has the added benefit of readability. It is easy to understand the reasoning process by looking at the agent's rule list. This feature will permit a better analysis of the results.

To be able to evolve this brain architecture we need to define its equivalent to the genome, and the operators that modify it on reproduction. The brain's genome is the rule list itself, no translation is applied. To alter it we defined only mutation operators. These operators are show in table 1.

Table 1: Mutation operators of the RuleList brain.

Operator	Description
<b>Mutate List</b>	This operator iterates through the rule list, and replaces a rule with a new random one.
<b>Mutate Rules</b>	This is the lowest level operator. It drills down to the perceptions on the conditions and mutates both the perceptions and their operators. It also mutates the action of the rules.
<b>Mutate Order</b>	This operator iterates through the rule list and moves a rule to the top of the list.
<b>Mutate Order 2</b>	This operator iterates through the rule list and moves a rule one position towards the top.

Not all of the mutation operators must be used. The programmer decides which of them to use for a particular experiment. In the case of the experiment described in this paper, the operators used were the *Mutate Order*, *Mutate Rules*, and *Mutate List*.

## Results

In this section we will expose and analyze the results of the experiments. But first we give an overview of the main configuration values used for the simulations. Most of the configuration values presented are the result of previous experimentation done on (Baptista and Costa, 2008).

The terrain is a square with sides of 1000 units. This field is populated initially with 20 agents, and that is also the minimum number of agents. These initial agents are generated with a brain composed of a random set of rules. The agents' brains are initialized with between 15 and 20 rules, having each up to 2 conditions. The initial and minimum number of food items is 200. One day lasts for 100 time units and the transitions from day to night, and vice versa, last 10 time units. The light level has a maximum of 100 and a minimum of 0, with a delta of 10. That means that for a given day the actual maximum light level will be between 100 and 90, and the minimum level between 0 and 10. Each agent is created with a vision range of 200 units and a vision angle of 60°. The agents reach is 20 units. Agents are initialized with energy 10 and consume a base metabolic rate of 0.1 energy units for each time unit. The metabolic rate increases by 0.01 when the agent is moving and drops by 0.03 when sleeping. The maximum age of the agents is 500 time units. The cost of reproduction is 2 energy units, plus the energy initialization for the child agent. Each food item gives an agent 3 energy units. The mutation probabilities are 0.01 for every operator. These experiments were run with a time limit of 200,000 time units. The caves were placed in the

environment at time 100,000.

Regarding the configuration parameters for the caves, as they represent the main addition to these experiments, we ran simulations with different configuration values. All caves have a side of 50. The field is populated with either 50, 20, or 10 caves.

For the experiments presented here, we ran 30 independent simulations for each configuration of the parameters. As is the case in (Baptista and Costa, 2008), from those 30 simulations, there are some where the agents will not be successful in evolving good foraging behavior, and thus will not be able to further evolve reproduction or synchronization to the day cycle. These experiments were not taken into account for some of the results shown in this section. Whenever that is the case, it will be stated.

Next, we will present and analyze the data of typical runs from the simulations. From all the simulations and runs analyzed, we found mainly two different types of plot (see figure 3 and figure 4). These represent the majority of results from the runs (except for those that are unsuccessful). Both the figures are taken from runs with the same parameter configuration (50 caves).

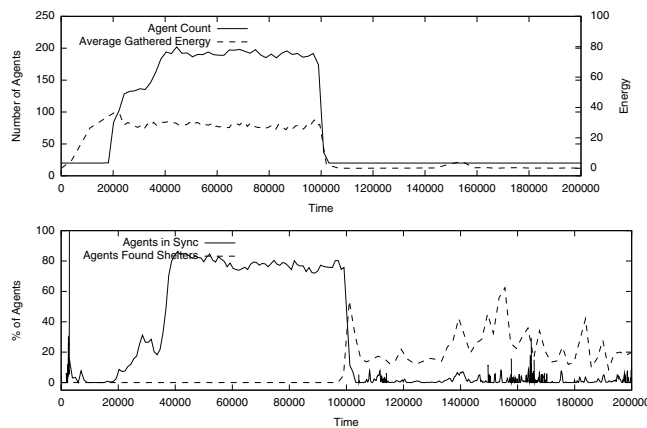


Figure 3: Plot of the evolution of the total number of agents in the population, average gathered energy, percentage of agents in sync with the day cycle, and percentage of agents that found caves, over the course of one simulation run (number of caves is 50).

The first type of plot, shown in figure 3, is what we expected to find in this experiment. Here, the population collapses and loses synchronization when the caves are inserted into the environment. Examining this plot, we can clearly see the agents are successful in evolving food gathering, reproduction, and synchronization in the first environment (up to time 100,000). Then, when the environment changes, most of the population dies and never recovers food gathering capabilities.

In fact, this data is consistent with what is presented in (Mirolli and Parisi, 2003). In that paper, the authors

show that, when the agents only have an input of the light level, they are not able to differentiate the “caves” from the “night”. They provide a possible solution to the problem, by incorporating in the structure of the brain, a clock source. However, in our simulations, we found that on a significant number of runs (see table 2) the agents do recover.

In figure 4 we show an example of a typical run where the agents recover in the second environment. In the first environment, we find a plot similar to that of figure 3. The agents develop good food gathering capabilities, reproduce, and synchronize with the daily cycle. When the environment changes at 100,000 time units, both the size of the population and the average gathered energy drop, but then quickly recover. More importantly, by analyzing the percentage of synchronized agents, we can see that, although it takes longer to recover, the agents also resynchronize to the daily cycle. One might wonder if it would be the case that the agents are simply not entering caves. But the plot also shows that about 80% of the agents find at least one cave during their lifetime. Note that, as these percentages are taken from the whole population at a given time interval, and there are constantly new agents being born, the percentage could never rise to 100%. New agents will normally need some time to move before they find a cave.

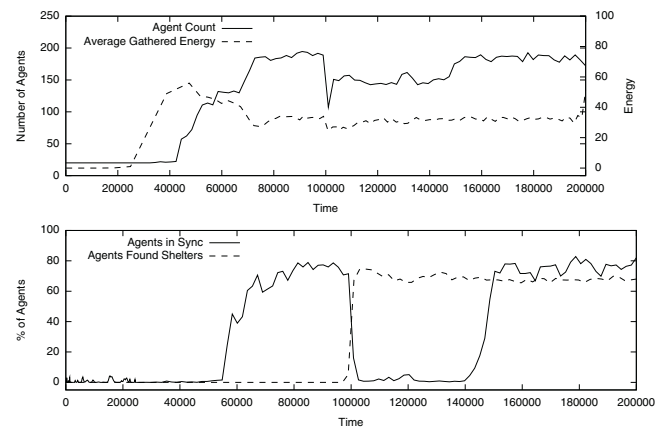


Figure 4: Plot of the evolution of the total number of agents in the population, average gathered energy, percentage of agents in sync with the day cycle, and percentage of agents that found caves, over the course of one simulation run (number of caves is 50).

These results may seem counter-intuitive, as there doesn't seem to be any way for the agents to detect the caves. To clarify, we analyzed some agents' brains. In listing 3 and listing 4 we show two examples of agents' brains from the simulation run shown in figure 4. The first one is taken from an agent living in the environment without caves (time 84536), and the second is taken from the environment with caves and at a time where the agents have resynchronized (time 183513).



**Listing 3** Example of the structure of the brain of an agent born at time 84536, for the simulation run shown in figure 4. Used Rules are set in bold.

1. **IF Resource Location = 3 THEN turn right**
2. IF Light Level < Light Level THEN eat
3. IF Resource Location > 3 THEN eat
4. **IF Light Level < 26.5859 THEN sleep**
5. IF Light Level < 36.3748 THEN sleep
6. IF Light Level = 47.8427 THEN eat
7. **IF Feature energy > 16.7159 THEN reproduce**
8. IF Feature energy = 26.0336 THEN reproduce
9. IF Resource Location = 3 THEN eat
10. IF Light Level < Feature energy THEN sleep
11. **IF istru(Reaching Resource) THEN eat**
12. **IF Resource Location < 2 THEN turn left**
13. IF istru(Reaching Resource) THEN turn left
14. **IF not(Reaching Resource) THEN go front**
15. IF Resource Location > 1 THEN sleep
16. IF istru(Reaching Resource) THEN turn right
17. IF Light Level < 97.9668 THEN turn right

The analysis of the brain in listing 3 allows us to see that the agent has good food gathering behavior (rules 1, 11, 12, and 14), reproduces whenever it has more than 16.7159 energy (rule 7), and sleeps when the light level falls below 26.5859 (rule 4). If put in the environment with caves, this agent would clearly not survive, as it would fall asleep on a cave (from rule 4) and never wake up again.

Examining the brain presented in listing 4, we can finally see how the agents adapt to the environment with caves. The important rule in this case is the one at position 2. To explain the behavior induced by this rule, we need to take a closer look. Lets first consider that the agent is in a cave. If the agent doesn't have any resource within its vision range (very likely as in the cave the vision range is small), the value of *Resource Location* will be 0. As we know, in a cave the light level is equal to the minimum light level, which is 0. Therefore, this rule will make the agent move forward whenever it is inside a cave. In fact, we find this rule (or some small variation) in all the agents analyzed in runs where there is recovery of synchronization.

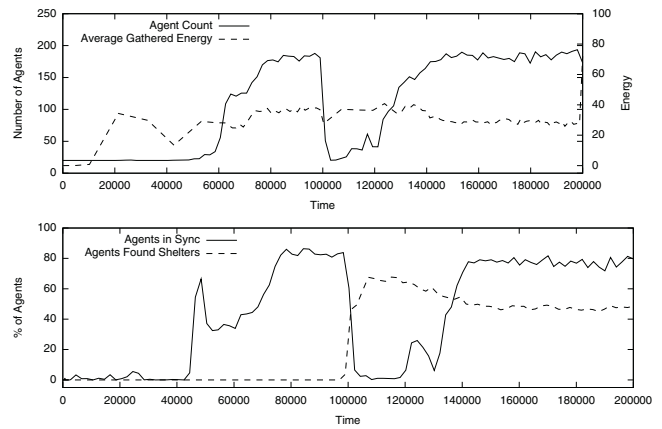


Figure 5: Plot of the evolution of the total number of agents in the population, average gathered energy, percentage of agents in sync with the day cycle, and percentage of agents that found caves, over the course of one simulation run (number of caves is 20).

The rest of the capabilities of the agent are also relatively easy to find in this rule list. From rules 1, 4, 13, and 14, we can see that the agent has a good foraging behavior. Rule 8 provides reproduction capabilities. And rule 16, when combined with rule 14, makes the agent sleep if the light level of the environment is less than 11.8449 and greater than 3 (maximum value for the *Resource Location* perception).

In table 2 we show an overview of the successful runs from all the tested simulation configurations. As stated earlier, we ran simulations with different values for the number of caves present in the environment. These results show that the configuration change doesn't seem to affect the number of successful runs out of the total of 30 runs. This was expected, as we were only changing the number of caves, which didn't affect the first environment. However, if we look at the number of runs where agents recover in the environment with caves, we find a different results for the three configurations. This is also rather straightforward to explain. With a smaller number of caves in the environment, the probability an agent has of finding a cave within its lifetime diminishes, making it easier maintain the synchronization from the first environment.

In figure 5 we show a run of the simulation with 20 caves. The plot is similar to that of figure 4, with the main difference being in the percentage of agents that find caves. In this case we can see that the percentage stabilizes at about 50%, whereas with the 50 cave configuration it stabilizes at about 80%. In the configuration with 10 caves, the percentage falls to about 30%. As expected, the less caves in the environment, the lower the probability of an agent finding a cave in its lifetime.

**Listing 4** Example of the structure of the brain of an agent born at time 183513, for the simulation run shown in figure 4. Used rules are set in bold

1. **IF istrue(Reaching Resource) THEN eat**
2. **IF Resource Location = Light Level THEN go front**
3. IF Light Level = 29.2361 THEN turn right
4. **IF Resource Location = 1 THEN turn left**
5. IF Resource Location < Resource Location THEN sleep
6. IF Feature energy < Feature energy THEN sleep
7. IF istrue(Reaching Resource) THEN eat
8. **IF Feature energy > 16.7159 THEN reproduce**
9. IF istrue(Reaching Resource) THEN reproduce
10. IF Light Level = Resource Location THEN eat
11. IF Light Level = 59.7254 THEN eat
12. IF istrue(Reaching Resource) THEN go front
13. **IF Resource Location = 2 THEN go front**
14. **IF Light Level > 11.8449 THEN turn right**
15. IF Resource Location < 0 THEN sleep
16. **IF Resource Location < Light Level THEN sleep**
17. IF Resource Location < 0 THEN reproduce

Table 2: Overview of the success of runs.

N. Caves	Runs	Successful	Don't Rec.	Recover
50	30	23	9	14
20	30	22	3	19
10	30	24	1	23

## Conclusion

Even though it seemed unlikely for the agents to distinguish “night” from “cave” with only the light level as an input perception, evolution found a way to use the “tools at hand” to solve the problem. In this case, the agents take advantage of a specific feature of the environment created. As the light level, when inside a cave, is always zero, and outside a cave, at night, it is between zero and ten, the agents adapted to that fact. It is important to note that the scenario was not designed with this in mind, and in that regard that behavior was unexpected. This result may have a parallel in the real world, where it is common to find species that take ad-

vantage of specific properties of the environment, creating niches.

We believe these results are mainly due to the open-ended nature of the model used. The inexistent explicit fitness function allows the modeler not to over-specify and guide the solution, giving more freedom to the evolutionary system to produce viable solutions. Also, when compared with a fixed structure neural network, the brain architecture used may provide some added flexibility, and allow for these unexpected behaviors to evolve.

Regarding future work, even though the simulations showed that the agents adapt to the new environment without an internal clock, it would still be interesting to incorporate a clock source in the architecture of the agents, and compare the results of the simulations.

With these experiments we continued our investigation into the evolution of complex behaviors through open-ended evolution simulations. Following the results from the previous simulations (Baptista and Costa, 2008), where we showed that using the model of the BitBang framework we are able to evolve complex behaviors, from random initial conditions, in an open-ended evolution environment, we now also show that by simply adding extra complexity to the simulated world, the agents continue to evolve new behaviors adapted to their new environmental conditions.

## References

- Baptista, T. and Costa, E. (2008). Evolution of a multi-agent system in a cyclical environment. *Theory in Biosciences*, 127(2):141–148.
- Baptista, T., Menezes, T., and Costa, E. (2006). Bitbang: A model and framework for complexity research. In *Proceedings of the European Conference on Complex Systems 2006*, Oxford, UK.
- Beckmann, B., McKinley, P., and Ofria, C. (2007). Evolution of an adaptive sleep response in digital organisms. *Proceedings of the 9th European conference on Advances in artificial life*, pages 233–242.
- Channon, A. (2000). Three evolvability requirements for open-ended evolution. In Maley, C. C. and Boudreau, E., editors, *Artificial Life VII Workshop Proceedings*, pages 39–40. Portland, OR.
- Mirolli, M. and Parisi, D. (2003). Artificial Organisms That Sleep. In Banzaf, W., Ziegler, J., Christaller, T., Dittrich, P., and Kim, J. T., editors, *Advances in Artificial Life: Proceedings of the 7th European Conference on Artificial Life, ECAL 2003*, pages 377–386. Springer Berlin / Heidelberg.
- Rand, D. A., Shulgin, B. V., Salazar, J. D., and Millar, A. J. (2006). Uncovering the design principles of circadian clocks: mathematical analysis of flexibility and evolutionary goals. *Journal of Theoretical Biology*, 238(3):616–635.
- Russell, S. and Norvig, P. (2002). *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, 2 edition.
- Strogatz, S. (2004). *Sync: The Emerging Science of Spontaneous Order*. Penguin Books, London.

# The Effect of the Environment in the Synthesis of Robotic Controllers: A Case Study in Multi-Robot Obstacle Avoidance using Distributed Particle Swarm Optimization

Ezequiel Di Mario, Iñaki Navarro and Alcherio Martinoli

Distributed Intelligent Systems and Algorithms Laboratory,  
School of Architecture, Civil and Environmental Engineering,  
École Polytechnique Fédérale de Lausanne  
{ezequiel.dimario, inaki.navarro, alcherio.martinoli}@epfl.ch

## Abstract

The ability to move in complex environments is a fundamental requirement for robots to be a part of our daily lives. While in simple environments it is usually straightforward for human designers to foresee the different conditions a robot will be exposed to, for more complex environments the human design of high-performing controllers becomes a challenging task, especially when the on-board resources of the robots are limited. In this article, we use a distributed implementation of Particle Swarm Optimization to design robotic controllers that are able to navigate around obstacles of different shape and size. We analyze how the behavior and performance of the controllers differ based on the environment where learning takes place, showing that different arenas lead to different avoidance behaviors. We also test the best controllers in environments not encountered during learning, both in simulation and with real robots, and show that no single learning environment is able to generate a behavior general and robust enough to succeed in all testing environments.

## Introduction

In simple environments, it is usually straightforward for human designers to anticipate the different conditions a robot will be exposed to. Thus, robotic controllers can be designed manually by simplifying the number of parameters or inputs used. However, for more complex environments, the human design of high-performing controllers becomes a challenging task. This is especially true if the on-board resources of the robot are limited, as humans may not be aware of how to exploit limited sensing capabilities.

Machine-learning techniques are an alternative to human design that can automatically synthesize robotic controllers in large search spaces, coping with discontinuities and nonlinearities, and find innovative solutions not foreseen by human designers. In particular, evaluative, on-board techniques can develop specific behaviors adapted to the environment where the robots are deployed.

The purpose of this paper is twofold. First, to verify whether different behaviors arise as a function of the learning environment in the adaptation of multi-robot obstacle avoidance. Secondly, to test how the learned behaviors perform in environments not encountered during learning, that

is, to evaluate how general are the solutions found in the learning process. The adaptation technique used is Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995), which allows a distributed implementation in each robot, speeding up the adaptation process and adding robustness to failure of individual robots.

The remainder of this article is organized as follows. Section Background introduces some related work on PSO, and on the influence of the environment in robotic adaptation. In the Hypotheses and Methods section we propose two hypotheses that motivate our research and describe the experimental methodology used to test them. Section Results and Discussion presents the experimental results obtained and discusses the validity of the proposed hypotheses. Finally, we conclude the paper with a summary of our findings and an outlook for our future work.

## Background

The background for this article is divided into two subsections, one briefly introducing PSO and related work on distributed implementations and robustness in the presence of noise, and the second one dealing with environmental complexity and its role in the adaptation of robotic controllers.

## Particle Swarm Optimization

PSO is a relatively new metaheuristic originally introduced by Kennedy and Eberhart (1995), which was inspired by the movement of flocks of birds and schools of fish. Because of its simplicity and versatility, PSO has been used in a wide range of applications such as antenna design, communication networks, finance, power systems, and scheduling. Within the robotics domain, popular topics are robotic search, path planning, and odor source localization (Poli, 2008).

PSO is well suited for distributed/decentralized implementation due to its distinct individual and social components and its use of the neighborhood concept. Most of the work on distributed implementation has been focused on benchmark functions running on computational clusters (Akat and Gazi, 2008; Rada-Vilela et al., 2011). Implemen-

tations with mobile robots are mostly applied to odor source localization (Turduv and Atas, 2010; Marques et al., 2006), and robotic search (Hereford and Siebold, 2007), where the particles' position is usually directly matched to the robots' position in the arena.

Most of the research on optimization in noisy environments has focused on evolutionary algorithms (Jin and Branke, 2005). The performance of PSO under noise has not been studied so extensively. Parsopoulos and Vrahatis (2001) showed that standard PSO was able to cope with noisy and continuously changing environments, and even suggested that noise may help to avoid local minima. Pan et al. (2006) proposed a hybrid PSO-Optimal Computing Budget Allocation (OCBA) technique for function optimization in noisy environments. Pugh and Martinoli (2009) showed that PSO could outperform Genetic Algorithms on benchmark functions and for certain scenarios of limited-time learning in the presence of noise.

In our previous work (Di Mario and Martinoli, 2012), we analyzed in simulation how different algorithmic parameters in a distributed implementation of PSO affect the total evaluation time and the resulting fitness. We proposed guidelines aiming at reducing the total evaluation time so that it is feasible to implement the adaptation process within the limits of the robots' energy autonomy.

### Role of the Environment

Regarding complexity, Al-Kazemi and Habib (2006) analyzed the internal behavior of PSO when the dimension of the problem is increased. They used different metrics to conclude that the PSO particles behave in a similar way independently of the complexity of the problem. Auerbach and Bongard (2012) studied the relationship between environmental and morphological complexity in evolved robots, showing that many complex environments lead to the evolution of more complex body forms than those of robots evolved in simple environments.

Nolfi (2005) proposed that the behavior of a robot (and of any other agent) depends on the interaction between its controller, its body, and the external environment (that can also consist of other robots). These interactions are non-linear and affect the behaviors as well as the learning process.

Nolfi and Parisi (1996) evolved neural network controllers for robotic exploration, switching between two different environments during the evolution process. They evolved two different neural networks: with and without the capability to learn how to behave in the environment where the robot is placed. Different behaviors resulted from evolution depending on whether learning was allowed and on the environment where the robots were tested.

Islam and Murase (2005) evolved a robotic controller for obstacle avoidance and used tools from chaos theory (return maps and Lyapunov exponents) to measure the complexity of the resulting behaviors in the learning environment and

other testing environments.

Nelson et al. (2003) evolved robotic controllers while increasing the complexity of the environments during evolution. They compared the resulting fitness and evolution process with evolution performed only in the most complex world.

Berlanga et al. (2002) studied a coevolutionary method for robot navigation where the initial positions of the robots used for evolving the controllers are also evolved. They evolved solutions for several environments (in most cases of similar complexity), and tested their fitness in the arena where each controller was evolved as well as in the remaining arenas. They did not find significant performance differences between the controllers, probably due to the similar complexity of the arenas used for learning.

## Hypotheses and Methods

This article discusses how the environment affects the adaptation of controllers for multi-robot obstacle avoidance using a distributed implementation of PSO. Robots navigate autonomously in the presence of other robots in square arenas with obstacles of different size and shape. We look at the different environments where learning takes place, analyze the resulting behaviors, and test how the controllers perform in the environments where they did not previously learn.

### Hypotheses

The experiments conducted in this paper are motivated by the following hypotheses regarding the influence of the environment in the adaptation of robotic controllers:

**Hypothesis 1** *Different environments lead to different behaviors of the adapted controllers. This might be specially significant for considerably different environments (e.g., empty arena vs. very narrow corridor).*

**Hypothesis 2** *Some learning environments may generate more robust controllers that perform better in situations not encountered during learning. This leads to the problem of choosing the correct environment (or set of environments) for the adaptation process in order to make the resulting controller robust to variations in the environment.*

### Fitness Function

We use a metric of performance based on the work of Floreano and Mondada (1996), which is present in several studies on learning obstacle avoidance (e.g., Lund and Miglino (1996), Pugh and Martinoli (2009), Palacios-Leyva et al. (2013), and our own previous work Di Mario and Martinoli (2012)). The fitness function consists of three factors, all normalized to the interval [0, 1]:



$$f = f_v \cdot (1 - \sqrt{f_i}) \cdot (1 - f_i) \quad (1)$$

$$f_v = \frac{1}{N_{eval}} \sum_{k=1}^{N_{eval}} \frac{\max\{v_{l,k} + v_{r,k}, 0\}}{2} \quad (2)$$

$$f_t = \frac{1}{N_{eval}} \sum_{k=1}^{N_{eval}} \frac{|v_{l,k} - v_{r,k}|}{2} \quad (3)$$

$$f_i = \frac{1}{N_{eval}} \sum_{k=1}^{N_{eval}} i_{max,k} \quad (4)$$

where  $\{v_{l,k}, v_{r,k}\}$  are the normalized speeds of the left and right wheels at time step  $k$ ,  $i_{max,k}$  is the normalized proximity sensor activation value of the most active sensor at time step  $k$ , and  $N_{eval}$  is the number of time steps in the evaluation period. This function rewards robots that move forwards quickly ( $f_v$ ), turn as little as possible ( $f_t$ ), and stay away from obstacles ( $f_i$ ).

### Experimental Platform

Our experimental platform is the Khepera III, a differential wheeled robot with a diameter of 12 cm. It is equipped with nine infra-red sensors for short range obstacle detection, which in our case are the only external inputs for the controllers, and two wheel encoders, which are used to measure the wheel speeds for the fitness calculations.

Since the response of the Khepera III proximity sensors is not a linear function of the distance to the obstacles, the proximity values are inverted and normalized using measurements of the real robot sensor's response as a function of distance. This inversion and normalization results in a proximity value of 1 when touching an obstacle, and a value of 0 when the distance to the obstacle is equal to or larger than 10 cm.

Simulations are performed in Webots (Michel, 2004), a realistic physics-based submicroscopic simulator that models dynamical effects such as friction and inertia. In this context, by submicroscopic we mean that it provides a higher level of detail than usual microscopic models, faithfully reproducing intra-robot modules (e.g., individual sensors and actuators).

### Controller Architecture

The controller architecture used is a recurrent artificial neural network of two units with sigmoidal activation functions  $s(\cdot)$ . The outputs of the units determine the wheel speeds  $\{v_{l,t}, v_{r,t}\}$ , as shown in Equation 5. Each neuron has 12 input connections: the 9 normalized infrared sensors values  $\{i_1, \dots, i_9\}$ , a connection to a constant bias speed, a recurrent connection from its own output, and a lateral connection from the other neuron's output, resulting in 24 weight parameters in total  $\{w_0, \dots, w_{23}\}$ .

$$\begin{aligned} v_{l,t} &= s(w_0 + \sum_{k=1}^9 i_k \cdot w_k + w_{10} \cdot v_{l,t-1} + w_{11} \cdot v_{r,t-1}) \\ v_{r,t} &= s(w_{12} + \sum_{k=1}^9 i_k \cdot w_{k+12} + w_{22} \cdot v_{l,t-1} + w_{23} \cdot v_{r,t-1}) \end{aligned} \quad (5)$$

### Environments

We conduct experiments in four different environments, shown in Figure 1. The first one is an empty square arena of 2 m x 2 m, where the walls and the other robots are the only obstacles. The second and third environments are based on the same bounded arena, where cylindrical obstacles of two sizes are added in different numbers. The second environment has 20 medium-sized obstacles (diameter 10 cm), while the third has 40 small-sized obstacles (diameter 2 cm). The fourth environment is the same size as the empty arena with an inner wall of 1.5 m creating a continuous corridor of 25 cm width.

In simulation, the cylindrical obstacles are randomly repositioned before each fitness evaluation, meaning that the second and third environments are dynamic. In real-robot experiments, the obstacles are kept in fixed positions, the variation between runs is provided by the randomized initial pose of the robots. The third environment was not tested with real robots given the difficulty of keeping such thin cylinders vertical during collisions, but it should be noted that this kind of obstacles can occur in real environments, for example in the case of a chair or table with very thin legs.

All experiments are conducted with 4 robots. The method for initializing the robots' pose for each fitness evaluation is different between simulation and experiments with real robots. In simulation, the initial positions are set randomly with a uniform probability distribution, verifying that they do not overlap with obstacles or other robots. For the experiments with real robots, in the empty arena a random speed is applied to each wheel for three seconds to randomize the robots' pose. In the two arenas with obstacles and in the corridor one, the robots are manually repositioned to avoid disturbing the location of the obstacles, and then the robots turn in place with a random speed for two seconds to randomize their orientation.

### Adaptation Algorithm

The optimization problem to be solved by the adaptation algorithm is to choose the set of weights  $\{w_0, \dots, w_{23}\}$  of the artificial neural network controller such that the fitness function  $f$  as defined in Equation 1 is maximized. The chosen algorithm is the distributed, noise-resistant variation of PSO introduced by Pugh and Martinoli (2009), which operates by re-evaluating personal best positions and aggregating them with the previous evaluations (in our case a regular

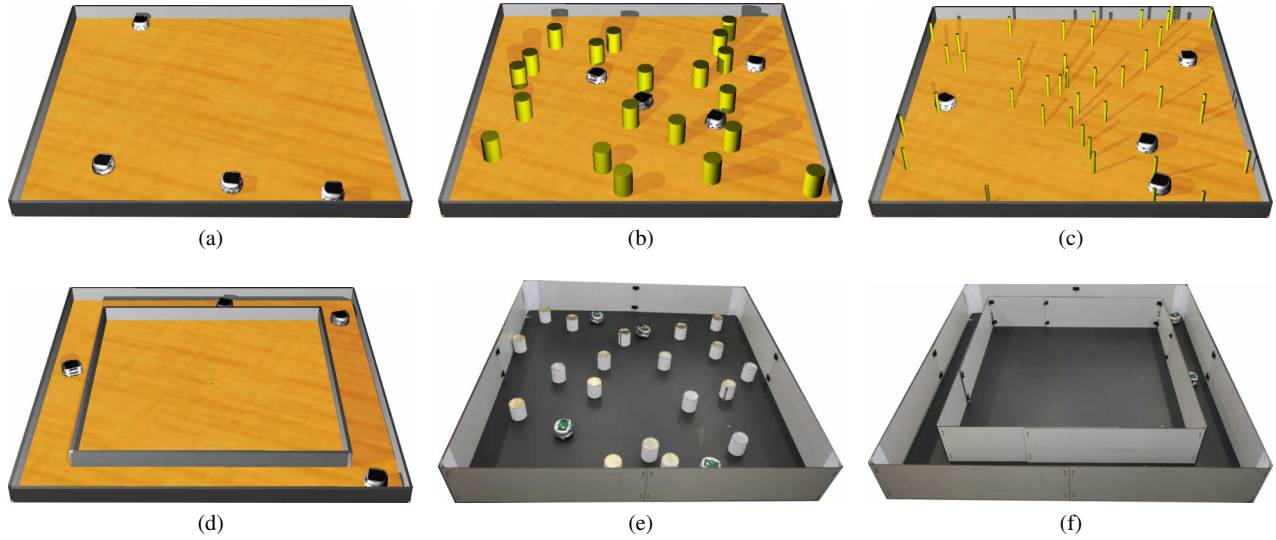


Figure 1: Different environments used in the adaptation and evaluation of the controllers. (a) Empty arena in simulation. (b) Medium-sized obstacles arena in simulation. (c) Small-sized obstacles arena in simulation. (d) Corridor arena in simulation. (e) Real medium-sized obstacles arena. (f) Real corridor arena.

---

```

1: Initialize particles
2: for  $N_i$  iterations do
3:   for  $\lceil N_p/N_{rob} \rceil$  particles do
4:     Update particle position
5:     Evaluate particle
6:     Re-evaluate personal best
7:     Aggregate with previous best
8:     Share personal best
9:   end for
10: end for

```

---

Figure 2: Noise-resistant PSO algorithm

average performed at each iteration of the algorithm). The pseudocode for the algorithm is shown in Figure 2.

The position of each particle is a 24-dimensional real-valued vector that represents the weights of the artificial neural network. The velocity of particle  $i$  in dimension  $j$  (shown in Equation 6) depends on three components: the velocity at the previous step weighted by an inertia coefficient  $w_I$ , a randomized attraction to its personal best  $x_{i,j}^*$  weighted by  $w_P$ , and a randomized attraction to the neighborhood's best  $x_{p',j}^*$  weighted by  $w_N$ .  $rand()$  is a random number drawn from a uniform distribution between 0 and 1. The position of each

particle is updated according to Equation 7.

$$v_{i,j} := w_I \cdot v_{i,j} + w_P \cdot rand() \cdot (x_{i,j}^* - x_{i,j}) + w_N \cdot rand() \cdot (x_{p',j}^* - x_{i,j}) \quad (6)$$

$$x_{i,j} := x_{i,j} + v_{i,j} \quad (7)$$

The algorithm is implemented in a distributed fashion, which reduces the total evaluation time required by a factor equal to the number of robots. Even if the learning in this paper is performed only in simulation, the algorithm can easily be executed completely on-board with very low requirements in terms of computation and communication.

Each robot evaluates in parallel a possible candidate solution and shares the solution with its neighbors in order to create the next pool of candidate solutions. The neighborhood presents a ring topology with one neighbor on each side. Particles' positions and velocities are initialized randomly with a uniform distribution in the  $[-20, 20]$  interval, and their maximum velocity is also limited to that interval.

The PSO algorithmic parameters are set following the guidelines for limited-time adaptation we presented in our previous work (Di Mario and Martinoli, 2012) and are shown in Table 1.

## Results and Discussion

The results of this article are presented as follows. First, we perform the learning in simulation in the four environments previously mentioned. Then, the best controller from each learning environment is tested in every environment in simulation. Finally, the four controllers from each learning

Table 1: PSO parameter values

Parameter	Value
Number of robots $N_{rob}$	4
Population size $N_p$	24
Iterations $N_i$	200
Evaluation span $t_e$	40 s
Re-evaluations $N_{re}$	1
Personal weight $w_p$	2.0
Neighborhood weight $w_N$	2.0
Dimension $D$	24
Inertia $w_I$	0.8
$V_{max}$	20

environment are also tested with real robots in three of the four environments.

### Learning in Simulation with PSO

Since PSO is a stochastic optimization method and the performance measurements are noisy, each PSO optimization run may converge to a different solution. Therefore, for statistical significance, we performed in simulation 100 PSO adaptation runs for each learning environment. Figure 3 shows the progress of the PSO learning at each iteration for the four environments. Vertical bars show the standard deviation among the 100 PSO runs.

The highest performance corresponds to the empty arena since it is the easiest environment with just the bounding walls and the other robots acting as obstacles. The fitness in both environments with cylindrical obstacles is very similar for the whole learning process. The slowest learning rate occurs for the narrow corridor, indicating that this environment is more challenging for the learning algorithm. By the end of the adaptation process the performance is slightly lower than in the arenas with cylindrical obstacles.

It should be noted that the learning environment has a significant impact in the variation between runs, as the standard deviation is lowest in the empty arena and it increases markedly for the more complex environments.

Trajectories can be a useful tool to identify the behavior of the robots, as we have seen in our previous work (Di Mario et al., 2011). Figure 4 shows the resulting trajectories of the best learned behaviors in simulation for each environment where adaptation took place. It can be seen how in the empty arena and in the medium-sized obstacles arena the robot trajectories are straight until they find an obstacle (wall, cylindrical obstacle, or other robot), performing then a sharp turn and continuing straight afterwards.

The trajectory learned in the arena with small-sized obstacles is curvilinear when there are no obstacles within range. When the robot detects an obstacle, it makes a sharp turn to later continue its curvilinear movement. The small obstacles are thinner than the distance between two contiguous infrared sensors, so sometimes the robots are not able to detect

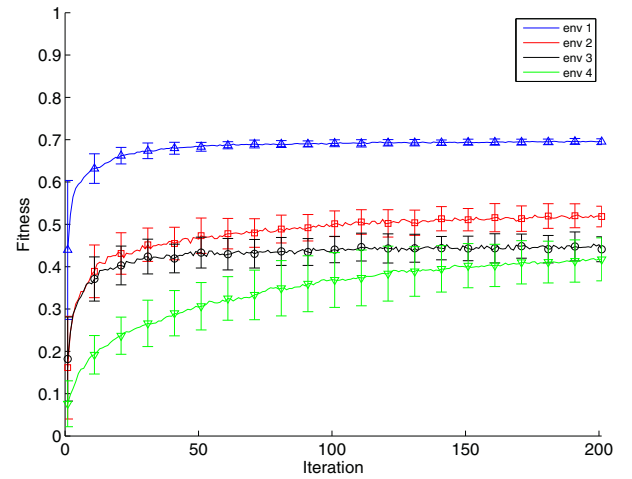


Figure 3: Best fitness found at each iteration for 100 PSO optimization runs. Bars represent the standard deviation across runs. Fitness in empty arena in blue (env 1). Fitness in arena with 20 medium cylindrical obstacles in red (env 2). Fitness in arena with 40 small cylindrical obstacles in black (env 3). Fitness in corridor arena in green (env 4).

them. Curvilinear movements may help in avoiding getting stuck in front of the small obstacles, and thus the behavior learned with PSO does not involve moving in straight lines as in the other cases.

In the corridor arena, the robot moves along the corridor, turning 90 degrees to head into the following sub-corridor, and thus exploring the whole arena.

As we conjectured in Hypothesis 1, the different environments cause the robots to learn different behaviors. In the next section we will show how the learned controllers behave in the other environments that were not encountered during learning.

### Testing in Simulation

In the previous section, we obtained four different controllers corresponding to each environment where learning took place. In this section, we test the controllers in all environments to see how they perform in situations not encountered while learning, i.e., to see how general and robust are the obtained behaviors.

Figure 5a shows the boxplot of the fitness of 20 evaluation runs performed in simulation for each controller and testing environment. Since all experiments are conducted with 4 robots, this results in 80 fitness measurements per controller and environment. For the sake of brevity, we use  $T$  to denote testing environment,  $L$  for learning environment, and we number the environments from one to four in the following order: empty arena, arena with 20 medium cylindrical obstacles, arena with 40 small cylindrical obstacles and

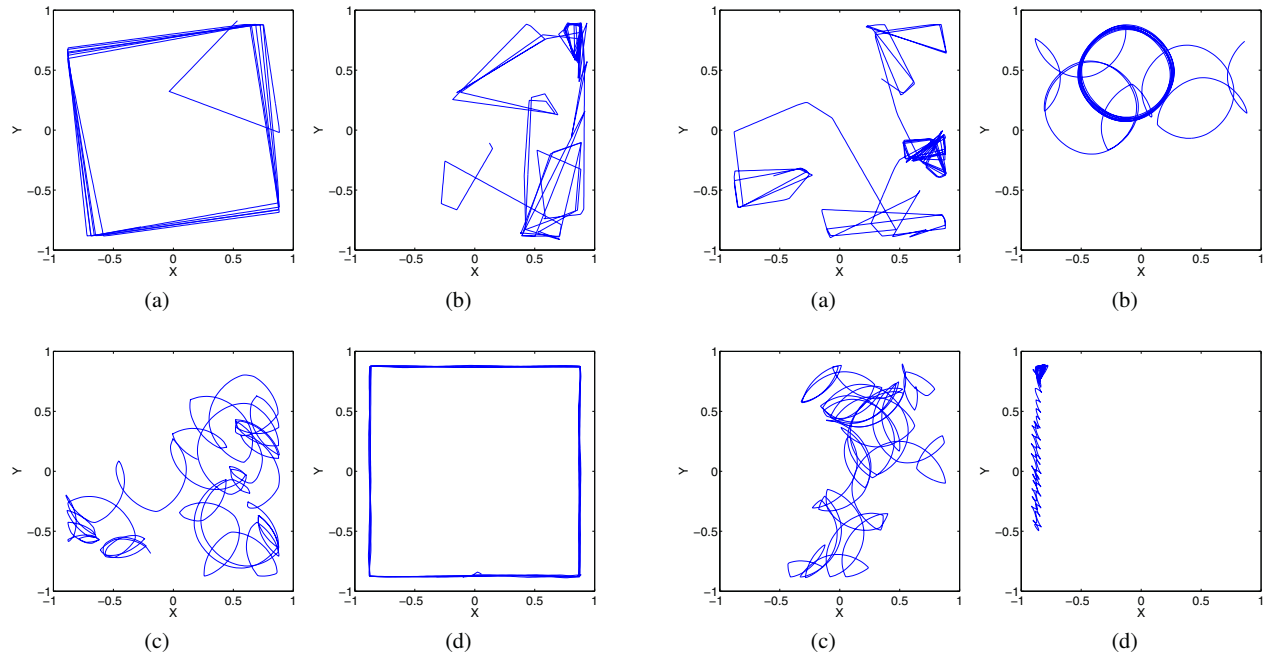


Figure 4: Trajectories of one of the four robots during a single experiment in simulation for the controllers learned in the four environments under study. (a) Empty arena. (b) Medium sized obstacles arena. (c) Small sized obstacles arena. (d) Corridor arena.

corridor arena. Thus,  $T1L4$  for instance should be read as: test performed in the empty environment with the controller learned in the corridor environment.

As expected, for each environment, the controller learned in the testing environment has the highest performance. However, for the simplest environment ( $T1$ ), there is no significant difference between the performance of controllers  $L1$ ,  $L2$ , and  $L4$ . Regarding Hypothesis 2 concerning the generality of the learned behaviors, controller  $L4$  seems to be the most robust, as it significantly outperforms all other controllers in the corridor and still performs almost as good as  $L1$  in  $T1$  and reasonable well in  $T2$ , although it performs poorly in  $T3$ .

Further insight on the performances can be obtained by analyzing the trajectories described by the robots in the different environments. Out of the 16 evaluation conditions, we show the ones we consider most interesting in Figure 6.

The behavior of controller  $L1$  is similar to that of controller  $L2$  in all testing environments (for example, compare the trajectories from Figure 6a and Figure 4b), since they employ similar avoidance strategies: moving in straight lines and making sharp turns near obstacles. This result becomes evident when considering that the medium-sized cylindrical obstacles are very similar in shape and size to

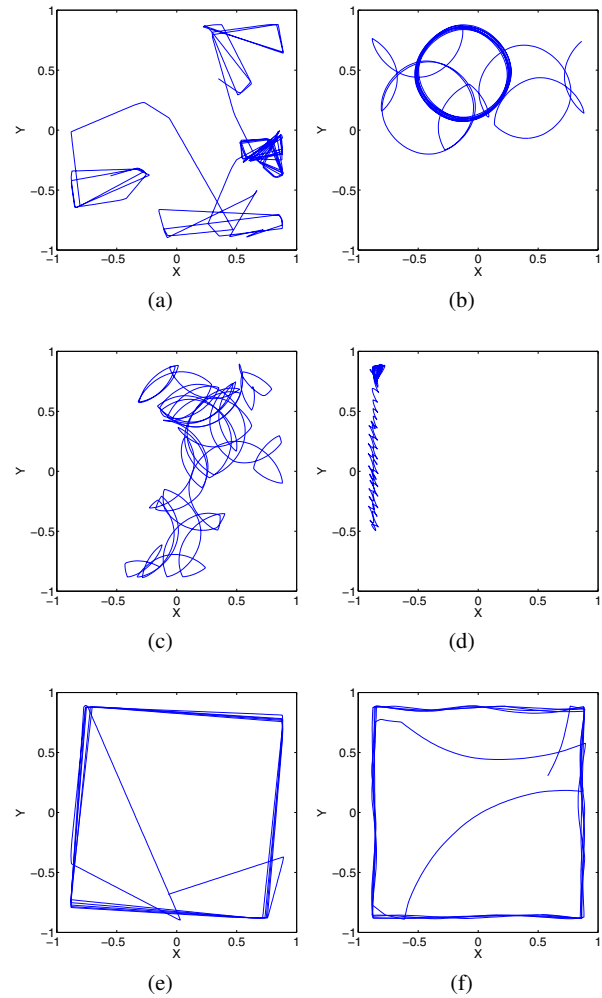


Figure 6: Trajectories of one of the four robots during a single experiment in simulation for different learned controllers (LX) and testing environments (TX). (a) T2L1. (b) T1L3. (c) T2L3. (d) T4L3. (e) T1L4. (f) T1L4\*.

the Khepera III robot. However, maybe due to the higher obstacle density of Environment 2, controller  $L2$  is more robust in the sense that it performs better in environments 3 and 4.

The curvilinear behavior of controller  $L3$ , which enables it to avoid very thin obstacles, is also observed with the larger obstacles of Environment 2 (Figure 6c), and results in fully circular trajectories in the empty environment (Figure 6b). However, this controller as well as controllers  $L1$  and  $L2$  were not able to move along the corridor, doing instead short straight movements alternated with sharp turns (Figure 6d).

Controller  $L4$  was the only one able to move smoothly along the corridor in Environment 4, performing well in all environments except  $T3$ . The behavior learned can be ob-



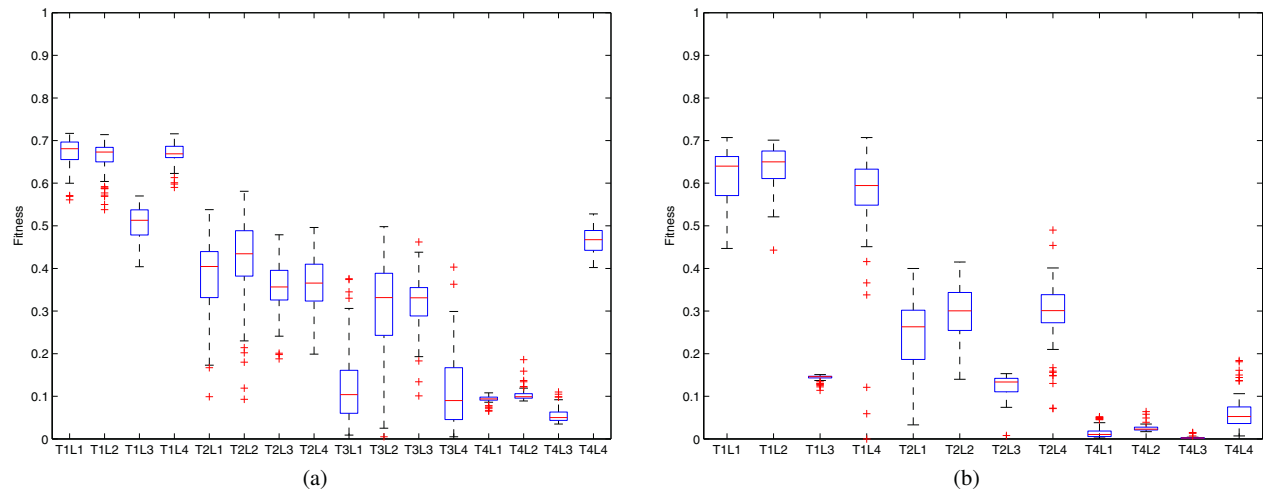


Figure 5: Boxplot showing the fitness of the four learned controllers (L1-L4). (a) Evaluated in the four testing environments (T1-T4) in simulation. (b) Evaluated in three testing environments (T1, T2 and T4) with real robots. The box represents the upper and lower quartiles, the line across the middle marks the median, and the crosses show outliers.

served when tested in the empty environment (*T1L4*) in Figure 6e. The robot moved straight performing a 90 degree sharp turn when finding an obstacle. This exact 90 degree turn was learned in the corridor environment to perform the transition from one sub-corridor to another.

As mentioned previously, we run 100 PSO runs for each environment, and controller *L4* is the best-performing one from the 100 runs in the corridor environment, but we noticed that not all the resulting controllers have the same behavior. A different controller resulting from the corridor environment is shown for the empty arena (*T1L4\**) in Figure 6f. The robot learned a wall-following behavior, performing a curvilinear movement in the absence of obstacles.

However, when testing this controller in the corridor (*T4L4\**) the trajectory looks exactly the same as the one from *T4L4*. Thus, it is interesting to notice that this behavior could only be observed when testing in other environments than the learning one, which shows the importance of using varied environments to observe the whole range of behaviors of a given controller.

### Testing with Real Robots

In order to validate the results obtained in simulation, we tested the same controllers with real robots in environments 1, 2, and 4. We did 20 evaluation runs with 4 robots, leading to 80 fitness measurements per case. The resulting fitness is shown in Figure 5b.

As in simulation, the performance of controllers *L1* and *L2* was similar. Again, controller *L4* seemed to be the most robust, outperforming all other controllers in the corridor and performing similarly to the best controllers in the other

two environments.

Controller *L3* suffered a noticeable performance drop when going from simulation to reality due to an unmodeled effect: the Khepera III motors' were not able to work smoothly at low speeds, and thus the inner wheel in the circular movements in open spaces was practically stopped, resulting in circles with a very small radius.

Finally, controller *L4* was also able to move along the corridor as in simulation, although the behavior was not as smooth and turns midway through the corridor were more frequent than in simulation (probably due to inaccuracies in the sensor model and the increased noise in real environments). Thus, the real-world performance was much lower.

### Conclusion

In this paper, we studied the effect of the environment on the multi-robot learning of an obstacle avoidance behavior. We showed that the same controller architecture, fitness function, and learning algorithm implemented in different environments lead to different avoidance behaviors, such as moving in straight lines with sharp turns, curvilinear movements, and wall-following around obstacles. We then tested the learned controllers in environments not encountered during learning, both in simulation and with real robots, which allowed us to see the full range of behaviors of each controller. Finally, we saw that no single learning environment was able to generate a behavior general enough to succeed in all testing environments.

As future work, we intend to study the interplay between architectural complexity and capability of generalization. In other words, we would like to know how to design a learn-

ing environment, or maybe a set of environments if required, that lead to general and robust avoidance behaviors while maintaining the architecture complexity low. It would also be interesting to study the interplay between a certain fitness function and the required architecture complexity. This work is part of our ongoing effort to develop distributed, noise-resistant adaptation techniques that can optimize high-performing robotic controllers quickly and robustly.

### Acknowledgement

This research was supported by the Swiss National Science Foundation through the National Center of Competence in Research Robotics.

### References

- Akat, S. B. and Gazi, V. (2008). Decentralized asynchronous particle swarm optimization. In *IEEE Swarm Intelligence Symposium*.
- Al-Kazemi, B. and Habib, S. (2006). Complexity analysis of problem-dimension using PSO. In *WSEAS International Conference on Evolutionary Computing*, pages 45–52.
- Auerbach, J. E. and Bongard, J. C. (2012). On the relationship between environmental and morphological complexity in evolved robots. In *Genetic and Evolutionary Computation Conference*, pages 521–528. ACM Press.
- Berlanga, A., Sanchis, A., Isasi, P., and Molina, J. M. (2002). Neural network controller against environment: A coevolutionary approach to generalize robot navigation behavior. *Journal of Intelligent and Robotics Systems*, 33(2):139–166.
- Di Mario, E. and Martinoli, A. (2012). Distributed Particle Swarm Optimization for Limited Time Adaptation in Autonomous Robots. In *International Symposium on Distributed Autonomous Robotic Systems 2012, Springer Tracts in Advanced Robotics 2014 (to appear)*. Available at <http://infoscience.epfl.ch/record/182403>.
- Di Mario, E., Mermoud, G., Mastrangeli, M., and Martinoli, A. (2011). A trajectory-based calibration method for stochastic motion models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4341–4347.
- Floreano, D. and Mondada, F. (1996). Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(3):396–407.
- Hereford, J. and Siebold, M. (2007). Using the particle swarm optimization algorithm for robotic search applications. In *IEEE Swarm Intelligence Symposium*, pages 53–59.
- Islam, M. M. and Murase, K. (2005). Chaotic dynamics of a behavior-based miniature mobile robot: effects of environment and control structure. *Neural Networks*, 18(2):123 – 144.
- Jin, Y. and Branke, J. (2005). Evolutionary Optimization in Uncertain Environments - A Survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317.
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *IEEE International Conference on Neural Networks*, pages 1942 – 1948 vol.4.
- Lund, H. and Miglino, O. (1996). From simulated to real robots. In *IEEE International Conference on Evolutionary Computation*, pages 362–365.
- Marques, L., Nunes, U., and Almeida, A. T. (2006). Particle swarm-based olfactory guided search. *Autonomous Robots*, 20(3):277–287.
- Michel, O. (2004). Webots: Professional Mobile Robot Simulation. *Advanced Robotic Systems*, 1(1):39–42.
- Nelson, A., Grant, E., Barlow, G., and White, M. (2003). Evolution of complex autonomous robot behaviors using competitive fitness. In *International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, pages 145–150.
- Nolfi, S. (2005). Behaviour as a complex adaptive system: On the role of self-organization in the development of individual and collective behaviour. *ComplexUs*, 2(3-4):195–203.
- Nolfi, S. and Parisi, D. (1996). Learning to adapt to changing environments in evolving neural networks. *Adaptive Behavior*, 5:75–98.
- Palacios-Leyva, R. E., Cruz-Alvarez, R., Montes-Gonzalez, F., and Rascon-Perez, L. (2013). Combination of reinforcement learning with evolution for automatically obtaining robot neural controllers. In *IEEE International Conference on Evolutionary Computation*, pages 119–126.
- Pan, H., Wang, L., and Liu, B. (2006). Particle swarm optimization for function optimization in noisy environment. *Applied Mathematics and Computation*, 181(2):908–919.
- Parsopoulos, K. E. and Vrahatis, M. N. (2001). Particle Swarm Optimizer in Noisy and Continuously Changing Environments. In Hamza, M. H., editor, *Artificial Intelligence and Soft Computing*, pages 289–294. IASTED/ACTA Press.
- Poli, R. (2008). Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications*, 2008(2):1–10.
- Pugh, J. and Martinoli, A. (2009). Distributed scalable multi-robot learning using particle swarm optimization. *Swarm Intelligence*, 3(3):203–222.
- Rada-Vilela, J., Zhang, M., and Seah, W. (2011). Random Asynchronous PSO. *The 5th International Conference on Automation, Robotics and Applications*, pages 220–225.
- Turduev, M. and Atas, Y. (2010). Cooperative Chemical Concentration Map Building Using Decentralized Asynchronous Particle Swarm Optimization Based Search by Mobile Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4175–4180.

# Multi-Objective Optimization of Intrusion Detection Systems for Wireless Sensor Networks

Martin Stehlík<sup>1</sup>, Adam Saleh<sup>1</sup>, Andriy Stetsko<sup>1</sup> and Vashek Matyáš<sup>1</sup>

<sup>1</sup>Masaryk University, Brno, Czech Republic  
xstehl2@fi.muni.cz

## Abstract

Intrusion detection is an essential mechanism to protect wireless sensor networks against internal attacks that are relatively easy and not expensive to mount in these networks. Recently, we proposed, implemented and tested a framework that helps a network operator to find a trade-off between detection accuracy and usage of resources that are usually highly constrained in wireless sensor networks. We used a single-objective optimization evolutionary algorithm for this purpose. This approach, however, has its limitations. In order to eliminate them, we show benefits of multi-objective evolutionary algorithms for intrusion detection parametrization and examine two multi-objective evolutionary algorithms (NSGA-II and SPEA2). Our examination focuses on the impact of an evolutionary algorithm (and its parameters) on the optimality of found solutions, the speed of convergence and the number of evaluations.

## Introduction

Recent advances in wireless communications and low-cost electronic devices enabled the development of *low-cost* and *high-performance wireless* networking technologies. Apart from widely used *cellular networks* known from mobile phones and *infrastructure local area networks*, there are also *ad hoc networks* operating without any given and fixed infrastructure, where the connections are established on demand in ad hoc manner (Dressler, 2007).

*Wireless sensor networks* (WSNs) can be considered as a type of ad hoc wireless networks with many specifics. The main difference against the “ordinary” ad hoc wireless networks is that the WSNs consist of a large number of usually *homogeneous, low-cost* and *resource restricted sensor nodes*. Their goal is to measure physical parameters like temperature, humidity, intensity of light, and send it to a *base station* (BS) for further processing. Since a node communication range is limited to tens of meters and it is not always feasible for the node to directly communicate with the BS, measurements are usually sent hop-by-hop from one node to another until they reach the BS.

Since WSNs are often deployed in physically open and sometimes even hostile environments, they can be subject to various security attacks ranging from *passive eavesdropping*

to *active* interfering (Zhang and Lee, 2000). An active attacker may insert a node in a network, or capture and reprogram an existing one in order to, e.g., drop, delay, modify or reorder packets containing important sensor measurements or routing information (Karlof and Wagner, 2003).

An *intrusion detection system* (IDS) is an essential mechanism to protect a network against internal attacks. Sensor nodes can monitor only a small part of their surrounding. Hence, to enable intrusion detection even in the remote parts of the network, *intrusion detection agents* should be deployed at different nodes in all parts of the network to monitor malicious events *locally*, in a distributed fashion (da Silva et al., 2005; Roman et al., 2006).

An IDS for a WSN should be highly optimized for a given application scenario, i.e., it should not consume more energy (memory) than it is necessary to achieve a required level of detection accuracy. Otherwise, a higher detection accuracy will be at the expense of resources that are highly constrained in sensor nodes. For example, MICAz – a typical sensor node – is equipped with the 8 MHz Atmel Atmega128L microcontroller, 4 kB RAM, 512 KB flash memory, 802.15.4 compliant Texas Instruments CC2420 transceiver and two AA batteries (Crossbow, 2013).

One can expect different requirements in network security and resource consumption for different applications, e.g., one for emergency-response and another for agriculture. Since there is a variety of application scenarios, a single set of IDS parameters is not optimal for all of them. In (Stetsko, 2012), a framework that optimizes the parameters of an IDS for a given application scenario in terms of the detection accuracy and resource consumption was presented. The optimization was driven by multiple objectives that were represented by different evaluation metrics (e.g., number of true positives, number of true negatives, memory usage). A single-objective optimization algorithm was used with the need to provide weights for each objective before the optimization process takes place. If the network operator wants to change weights, the optimization process should run again.

In this work, we examine two multi-objective evolution-

ary algorithms (MOEAs) that eliminate the limitation mentioned above. Our contribution is threefold. First, we show that MOEAs can be utilized for optimization of an IDS in WSNs. Second, we compare effectiveness of NSGA-II and SPEA2 for our scenario that is described in the following sections. Third, we evaluate the impact of MOEA parameters on the optimization process.

## Optimization Framework for WSN

In this section, we present our optimization framework that can be used for optimization of various detection techniques in WSNs. It consists of a *simulator* that simulates a specific configurable scenario of a WSN and provides statistics of the simulation. The statistics are input for an *optimization engine* that, based on them, designs new WSN configurations and provides them back to the simulator for further evaluation. The conceptual architecture of the framework is discussed in details in (Stetsko, 2012). In the following subsections, we present our use case that is consistent for all experiments discussed farther in this paper.

### Simulator

We evaluate the performance of the IDS using the MiXiM simulator (Köpke et al., 2008) based on the OMNeT++ platform. In the past, we made a thorough comparison of available simulators for WSNs in (Stetsko et al., 2011). The MiXiM simulator provides complex and realistic models suitable for our research. The simulation models are following: *wireless channel* and *network topology* models regarding the global aspects of the WSN and models of *network*, *data link* and *physical* layers and *energy consumption* models of the sensor nodes.

In our case, we simulate a WSN consisting of sensor nodes equipped with the CC2420 transceiver (widely used by MICAz and TelosB platforms) in an open environment. The description of settings of different simulation models follows:

- *Wireless channel model* – An open changing environment is simulated using the *log-normal shadowing* model (Rapaport, 2001) that is the most widely used wireless channel model among the simulators (Stetsko et al., 2011). The pass loss exponent representing the signal propagation was set up to 2 (outdoor environment). The variations in received signal are reflected by a Gaussian random variable with zero mean and standard deviation set up to 2. The time interval of the changes was set up to 0.001 s.
- *Network topology model* – Static topology with random uniform distribution of the sensor nodes in 2D square area is used in the experiments.
- *Network layer* – The network layer uses static routing tree generated using the following algorithm. A base station

broadcasts a packet containing its identification together with the value  $h$  (number of hops to the base station) set to 0. A node waits until it receives a packet from a neighbour that is the closest one (has the highest signal strength). Then the node sets the neighbour as its parent, increases value  $h$  by 1 and broadcasts the value together with its identification.

- *Data link layer* – Protocol CSMA-CA according to the IEEE 802.15.4 standard is used.
- *Physical layer* – The radio model represents the CC2420 transceiver that is compliant to the IEEE 802.15.4 standard and is used by MICAz and TelosB sensor nodes. The transmitting power is set up to -25 dBm (0.00316227766017 mW) for all sensor nodes.
- *Energy consumption* – The energy consumption is not taken into account in this paper because we do not use any sleep mode of the nodes' transceivers that saves the energy in presented detection technique.

### Optimization Engine

Various metaheuristics can be used to generate new candidate IDS configurations based on the previous ones evaluated by the simulator. We use *evolutionary algorithms* (EAs) as we found them advantageous in our previous work (Stetsko, 2012). The new generation of candidate configurations is evaluated by the simulator and the process continues until some stopping criterion is fulfilled (in our case predefined number of generations). In this paper, we show how the *multi-objective* approach can be utilized for IDS optimization in WSN. The MOEAs are discussed in section "Optimization Using MOEA". For experiments in this paper, we use a framework for metaheuristics ParadisEO (INRIA Lille, 2013; Liefvooghe et al., 2011).

### Distributed Computation

Apart from optimization using MOEAs, we decided to perform also *exhaustive search* for all our experiments. Since it is computationally demanding, we use the BOINC distributed computing platform (Anderson, 2001). We expect the network operators to optimize even more complex scenarios, where the exhaustive search would be unfeasible. However, to allow for a thorough comparison of the MOEAs, as one of the goals of this paper, we precomputed all configurations using BOINC on several tens of CPUs.

### Intrusion Detection System

In our optimization scenario, we use a detection technique the goal of which is to reveal malicious sensor nodes that execute *selective forwarding attack* where the attacker forwards only a fraction of received packets (Karlof and Wagner, 2003). In this kind of attack, it is assumed that the traffic is routed also through these malicious nodes. These nodes



are supposed to forward the packets received from their children to their parents towards the BS. The intent of the malicious nodes is to filter the traffic and forward only some packets which are selected randomly or based on some criteria (e.g., based on the data content of the packets to suppress information on some specific event in the environment).

### Detection Technique

We use a simple but *configurable* technique for detection of selective forwarding attack. Following notations are used in the text to explain the functionality of our IDS:

**Notation 1** The set  $A = \{a_1, \dots, a_{n_m}\}$  is a set of all malicious nodes in a network.

**Notation 2** The set  $C = \{c_1, \dots, c_{n_b}\}$  is a set of all benign nodes in a network.

**Notation 3** The function  $x : \mathbb{N} \rightarrow \mathbb{N}$  takes a sensor node index as an argument, and returns a number of the neighbours that consider this node benign.

**Notation 4** The function  $y : \mathbb{N} \rightarrow \mathbb{N}$  takes a sensor node index as an argument, and returns a number of the neighbours that consider this node malicious.

**Notation 5** The function  $n : \mathbb{N} \rightarrow \mathbb{N}$  takes a sensor node index as an argument, and returns a number of the neighbours of this node.

**Notation 6** The function  $m : \mathbb{N} \rightarrow \mathbb{N}$  takes a sensor node index as an argument, and returns the amount of memory (in bytes) used by an IDS on this node.

An IDS is running on a sensor node and continuously analyzing sent and overheard packets. A monitoring node  $c_i \in C$  overhears to some extent both incoming and outgoing packets of all close enough monitored neighbours  $b_j \in C \cup A$ . Note that the set of monitored neighbours is a subset of all neighbours limited to  $p_1$  (*max monitored nodes*). Neighbour of the node  $c_i$  is every node  $b_k \in C \cup A$ , such that  $c_i$  overheard at least one packet from  $b_k$  during the simulation. An IDS stores a table, where each of  $p_1$  rows corresponds to a certain monitored node. The table contains the number of packets received (PR) and forwarded (PF) by a monitored node.

If the IDS on a node  $c_i$  overhears a packet P sent to a monitored node  $b_j$  and  $b_j$  should forward the packet (e.g.,  $b_j$  is not a base station), then the IDS stores P in the buffer and increments the PR counter of the monitored node  $b_j$ . The number of buffered packets is limited by  $p_2$  (*buffer size*). If a new packet arrives but the buffer is full, the oldest packet is removed from the buffer. When the IDS overhears the packet P being forwarded by the node  $b_j$ , it removes P from the buffer (if it is still there) and increments the PF counter of the node  $b_j$ . Since both the table and the buffer are limited by parameters  $p_1$  and  $p_2$ , respectively, the IDS monitors only the closest nodes and the latest packets.

The detection is done at the end of the simulation, based on the collected statistics. The node  $c_i$  considers the node  $b_j$  as a selective forwarder if the dropping ratio of  $b_j$ , i.e., ratio of a number of packets dropped to a number of packets received, is higher than  $p_4$  (*detection threshold*). If the node  $c_i$  overheard less than  $p_3$  (*min received packets*) packets received by the node  $b_j$  during the simulation, the node  $b_j$  cannot be considered malicious by the node  $c_i$  because the number of overheard packets is small and there is a high level of uncertainty. In this case, the node  $b_j$  is considered benign. Note that the node  $c_i$  considers the neighbour node  $b_k$  benign if it is not a *monitored neighbour*. To summarize it, the detection decision is based on the following conditions:

- A node  $b_j$  is considered *malicious* by the neighbour node  $c_i$  if node  $b_j$  is the *monitored neighbour* and the observed dropping ratio is higher or equal to the *detection threshold* and the node  $c_i$  overheard at least *min received packets* addressed to the node  $b_j$ .
- A node  $b_j$  is considered *benign* by the neighbour node  $c_i$  if node  $b_j$  is not the *monitored neighbour* or if the dropping ratio is lower than the *detection threshold* or the IDS overheard less than *min received packets* addressed to the node  $b_j$ .

The IDS parameters that we optimize are shown in Table 1. The number of minimum received packets is in the range of  $\langle 1, 100 \rangle$  and detection threshold can be set from all packets dropped to all packets forwarded. The other parameters are discussed in the following subsection.

Name	Description	Range	Step
$p1$	<i>Max monitored nodes</i>	$\langle 1, 50 \rangle$	1
$p2$	<i>Buffer size</i>	$\langle 1, 50 \rangle$	1
$p3$	<i>Min received packets</i>	$\langle 1, 100 \rangle$	1
$p4$	<i>Detection threshold</i>	$\langle 0.01, 1 \rangle$	0.01

Table 1: The list of IDS parameters.

### Memory Consumption

The detection accuracy is influenced by the amount of memory allocated for the IDS. Each sensor node  $j$  requires the following amount of memory (in bytes) for the IDS:

$$m(j) = 8 * p_1 + 16 * p_2, \quad (1)$$

where 8 bytes are required for every monitored neighbour (4 B for node ID, 2 B for PR counter and 2 B for PF counter) and 16 bytes are required for one slot in the buffer (4 B for source address, 4 B for receiver address, 4 B for destination address in a case of multiple BSs in the WSN and 4 B for unique ID of a packet).

We determined the upper bound for the number of nodes being monitored by an IDS agent to 50. Table of neighbours can occupy 400 B ( $50 * 8 \text{ B} = 400 \text{ B}$ ) at maximum. That would be already 10% of MicaZ RAM (4 kB). Additional memory is needed for the buffered packets. We set the upper bound of the buffer size to 50 because the proof-of-concept experiments showed that bigger sizes did not influence the IDS accuracy. Thus,  $50 * 16 \text{ B} = 800 \text{ B}$  can be allocated for the buffer at maximum that is 20% of the RAM.

## Optimization Using MOEA

In this section, we show how MOEAs can be utilized for optimization of intrusion detection systems in complex environments of wireless sensor networks.

MOEAs can be useful for optimization of IDSs and other aspects in WSNs, providing the network operators with a set of *non-dominated* solutions. Then the network operator can choose between, e.g., an optimized solution  $A$  with better IDS accuracy at the cost of higher memory consumption and another optimized solution  $B$  with lower memory consumption at the cost of worse IDS accuracy. The set of non-dominated optimal solutions is called *Pareto front* (Talbi, 2009) and the goal of the MOEAs is to find a good approximation of the true Pareto front.

Several MOEAs producing approximations of the true Pareto front have been proposed. However, a good MOEA should produce solutions close to the true Pareto front with high *diversity* and should *converge* in a relatively short time. The *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) proposed in (Deb et al., 2002) and the *Strength Pareto Evolutionary Algorithm 2* (SPEA2) proposed in (Zitzler et al., 2001) belong to the most widely used MOEAs (Talbi, 2009) and we compare them in this paper. Both algorithms are implemented in the evolutionary framework ParadisEO and also in MOGALib (MOGALib, 2013).

NSGA-II features two main criteria to provide good *convergence* and *diversification*, respectively: 1) *ranking using non-dominance concept* to sort the solutions according to the number of other solutions they are dominated by; and 2) *crowding distance* to keep the solutions spread as far from each other as possible.

The fitness values calculated for the solutions found by SPEA2 are based on the number of dominating solutions and their strength of dominance (to achieve *convergence*) and on the density estimation function (to achieve *diversification*).

## Objective Space

In the optimization of our IDS, we consider three following objective functions: number of false positives  $fp$ , number of false negatives  $fn$ , amount of consumed memory  $mem$ .

**Objective function 1.** The number of *false negatives* ( $fn$ ) of a solution  $x$  is calculated as follows:

$$fn(x) = \frac{1}{|A|} * \sum_{a_i \in A} \frac{x(a_i)}{n(a_i)}. \quad (2)$$

The values of  $fn$  range from 0 to 1. If every malicious node in the network is correctly detected by all of its neighbours,  $fn$  is equal to 0 and if none of malicious nodes is detected by any of its neighbours,  $fn$  is equal to 1.

**Objective function 2.** The number of *false positives* ( $fp$ ) of a solution  $x$  is calculated as follows:

$$fp(x) = \frac{1}{|C|} * \sum_{c_i \in C} \frac{y(c_i)}{n(c_i)}. \quad (3)$$

The values of  $fp$  range from 0 to 1. If every benign node in the network is considered benign by all of its neighbours,  $fp$  is equal to 0 and if all benign nodes are considered malicious by all of its neighbours,  $fp$  equals to 1.

**Objective function 3.** The consumed *memory* ( $mem$ ) in a solution  $x$  is averaged over all benign nodes in the WSN as follows:

$$mem(x) = \frac{1}{|C|} * \sum_{c_i \in C} m(c_i), \quad (4)$$

where  $m(c_i)$  is calculated using formula 1.

The values of  $mem$  range from 0, where the IDS is potentially switched off, to  $8 * p_1 + 16 * p_2$  that is 1200 bytes for our upper bounds of  $p_1 = 50$  and  $p_2 = 50$ .

All three objectives are *minimized*.

## Pareto Front Discussion

The true Pareto front of a *sparse topology* (discussed in the next section) found by exhaustive search is shown in Fig. 1. The goal of MOEA is to find IDS configurations that are close to the points of the true Pareto front as much as possible. In Fig. 1 (a), the view of the whole objective space is shown from the perspective of  $mem$ ,  $fp$  and  $fn$ . Fig. 1 (b) depicts the trade-off between  $fp$  and  $fn$ , where the resulting values depend on the *detection threshold* of the IDS (parameter  $p_4$ ). Fig. 1 (c) shows a slight increase of  $fp$  (see scale) with higher number of *monitored nodes* (parameter  $p_1$  influencing  $mem$ ). The  $fp$  increase is caused by the fact that an IDS monitors a higher number of legitimate neighbours, each of which may be falsely considered malicious. Finally, Fig. 1 (d) shows a rapid decrease of  $fn$  with a higher number of *monitored nodes* (parameter  $p_1$ ) that is caused by the fact that a higher number of neighbours (and hence malicious nodes) is monitored (detected).

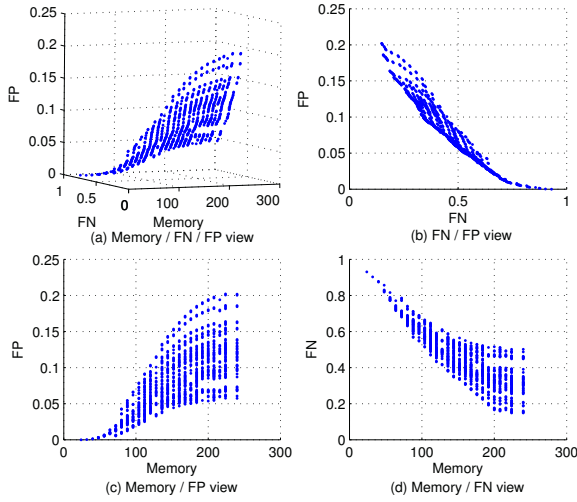


Figure 1: Optimal solutions on the Pareto front in our *objective space*.

### Comparison Methodology

In this section, we present a methodology that we used to compare NSGA-II and SPEA2 using multiple settings. We tried different configurations (see Table 2) of the algorithms on our problem and compare the outcome according to four metrics that are defined further in the text.

#### Evolution Parameters

The evolutionary algorithms have several parameters influencing the evolution process. The setting of the parameters is presented in this subsection.

**Initial population.** The initial population consists of randomly generated individuals (more specifically, the values of the IDS parameters are generated randomly within the predefined range of values).

**Population size.** The size of the population  $PopSize$  is set to following values:  $PopSize \in \{50, 100, 200\}$ .

**Crossover.** The multi-point crossover operation is applied with the probability  $PCross \in \{0.01, 0.1, 0.25, 0.5\}$ .

**Mutation.** The mutation operation is applied with probability  $PMut \in \{0.01, 0.1, 0.25, 0.5\}$  to every parameter  $p_1, \dots, p_4$  separately. When applying mutation, the parameter is changed randomly within an interval around the previous value of that parameter covering 10% of the overall parameter range (5% in both directions).

**Number of generations.** The number of generations  $NGen$  is set to  $NGen = 200$ .

Having the setting of the evolutionary parameters specified above, the IDS parameters were optimized with

$|PopSize| * |PCross| * |PMut| = 3 * 4 * 4 = 48$  different settings of both algorithms NSGA-II and SPEA2.

#### Metrics

Following notations are used to define metrics for the comparison of the algorithms:

**Notation 7** The set  $P$  is a set of all vectors (solutions) on the true Pareto front,  $\mathbf{p} \in P$  is a Pareto optimal vector and  $p = |P|$  is a size of the set  $P$ .

**Notation 8** The set  $N$  is a set of all vectors (solutions) found by the evolution,  $\mathbf{n} \in N$  is a non-dominated vector found by the evolution and  $n = |N|$  is a size of the set  $N$ .

Since we calculate the Euclidean distance between the solutions in the objective space in Metric 2 and Metric 3 defined below, we “normalize” the amount of the consumed memory in the following way:

**Notation 9**  $memn(x) = mem(x)/1200$ , where 1200 is the maximal amount of consumed memory in bytes and the range of the function  $memn$  is  $< 0, 1 >$ .

The MOEAs maintain an *archive* of the found non-dominated solutions. Two following aspects are expected of the solutions kept in the archive:

- *Convergence* – The approximation of the Pareto front should converge to the true Pareto front with new generations.
- *Diversification* – The found solutions should be uniformly distributed in the objective space in ideal case.

To be able to measure the effectiveness of the differently set MOEAs with respect to the aforementioned performance aspects, we use the following metrics to measure *convergence*:

**Metric 1.** The value of  $M_1$  is the number  $nd$  of non-dominated solutions  $\mathbf{nd} \in P$  found by the MOEA. The complexity of the calculation of this metric is  $O(nd * p)$ . This metric is used to measure *convergence*.

**Metric 2.** The value of  $M_2$  (*generational distance metric*) is the average of Euclidean distances from all found solutions  $\mathbf{n} \in N$  to the nearest solution  $\mathbf{p} \in P$  on the true Pareto front (Deb et al., 2002). This metric is also used to measure *convergence*. Having  $n$  solutions and  $p$  Pareto dominant solutions, the complexity of the calculation of this metric is  $O(n * p)$ .

Several metrics can be used to measure *diversification*, yet with an assumption that it is straightforward to find a neighbouring solution in the objective space. However, the definition of the neighbouring solution is easy for a two-dimensional objective space, but much more complicated for three- or more-dimensional objective spaces. In our case,

we have three objective functions. Thus, we use the following metric for diversification specified by Schott in (Schott, 1995) as *Spacing metric*. Note that the calculation of this metric does not require the set of the true Pareto front  $P$  found in the exhaustive search:

**Metric 3.** *Diversification* is measured using  $M_3$  as follows:  $M_3 = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2}$ , where  $d_i = \min_j \{|fn(i) - fn(j)| + |fp(i) - fp(j)| + |memn(i) - memn(j)|\}$  and  $\bar{d}$  is an average of all distances  $d_i, i \in \{1, \dots, n\}$ . If  $M_3 = 0$ , the solutions are spaced equally in the objective space. Having  $n$  solutions, the complexity of the calculation of this metric is  $O(n^2)$ .

We also calculate the time requirements of the MOEA, where the runtime of one simulation needed to evaluate a single individual in a population is the most time consuming element (around 8 minutes for our *Sparse topology* discussed farther). Note that the overall time is also dependent on the number of available CPUs:

**Metric 4.**  $M_4$  is the number of simulations needed in the whole evolution process.

We compare the number of individual evaluations during MOEA with number of evaluations needed for the exhaustive search for each of the experiments. In the exhaustive search, we evaluated 25,000,000 IDS configurations using our BOINC distributed computation platform.

### Our Test Case

We evaluated the performance of the NSGA-II and SPEA2 in two following different simulation scenarios consisting of 250 sensor nodes and 1 BS in 1) *sparse* and 2) *dense* topology. In both topologies, the goal of the IDSs was to detect five malicious nodes. Their placement can be found in (Stehlik et al., 2013).

- **Sparse topology** (Topology #1) – The sensor nodes are placed in the area of 200 x 200 m. The average area for one node is  $160 \text{ m}^2$ , i.e., the distance between two nearest neighbours is 12.65 m in average.
- **Dense topology** (Topology #2) – The sensor nodes are placed in the area of 100 x 100 m. The average area for one node is  $40 \text{ m}^2$ , i.e., the distance between two nearest neighbours is 6.33 m in average.

## Results and Discussion

In this section, we discuss the results of NSGA-II and SPEA2 configured in different ways. Evolutionary algorithm is a *stochastic* process – it means that multiple runs should be done for each configuration to get average behaviour of the algorithm. We ran the evolution 10 times for all the settings and we provide the *average* value  $Avg_x$  and

*standard deviation*  $\sigma_x$  for every metric  $M_x$  computed from results obtained for all ten evolution runs. Since we are limited in space, detailed results can be found in (Stehlik et al., 2013), as well as IDS settings of Pareto optimal solutions.

Table 2 shows how the results of the metrics using different set of evolution parameters are numbered in the charts presented in this section. MOEA settings used for sets 1, ..., 16 (*PopSize* = 50) are presented in the Table 2. Sets No. 17-32 and 33-48 have analogous MOEA settings for *PopSize* = 100 and *PopSize* = 200, respectively.

No.	1	2	3	4	5	6	7	8
<i>PCross</i>	0.01	0.01	0.01	0.01	0.1	0.1	0.1	0.1
<i>PMut</i>	0.01	0.1	0.25	0.5	0.01	0.1	0.25	0.5

No.	9	10	11	12	13	14	15	16
<i>PCross</i>	0.25	0.25	0.25	0.25	0.5	0.5	0.5	0.5
<i>PMut</i>	0.01	0.1	0.25	0.5	0.01	0.1	0.25	0.5

Table 2: The MOEA parameters settings for *PopSize* = 50.

### Sparse Topology

The lowest number of neighbours is characteristic for the *sparse topology*. A node  $b_j \in C \cup A$  has 41 neighbours in average. Hence, a lower amount of memory is needed to achieve IDS accuracy comparable to that in *dense topology*.

**Exhaustive Search** The true Pareto front obtained for the *sparse topology* is shown in Fig. 1. The results of the exhaustive search showed that 2,340 IDS configurations (out of 25,000,000) are Pareto dominant resulting in 996 unique solutions in the objective space (some IDS configurations have same results).

**Convergence** Fig. 2 shows solutions found by a single run of the MOEAs for set No. 45 that provided good results in a relatively short time. We consider a case where NSGA-II found 121 mutually non-dominated solutions, where 29 were Pareto optimal. SPEA2 found 80 mutually non-dominated solutions, where 20 were Pareto optimal. Objective space for evolution set No. 48, where solutions found by SPEA2 are spread better, can be found in (Stehlik et al., 2013).

In Fig. 3 (a) and (b), the impact of the evolution parameters on the convergence is shown. NSGA-II found more solutions on the Pareto front (metric  $M_1$ ) in most cases than SPEA2. Nevertheless, SPEA2 has better results measured by metric  $M_2$ . This is caused by NSGA-II solutions that have redundant amount of consumed memory remained in the population archive (e.g., solutions on the right-top corner in Fig. 2). These solutions were not dominated by other solutions with lower *fn*, *fp* and *mem* during the evolution.



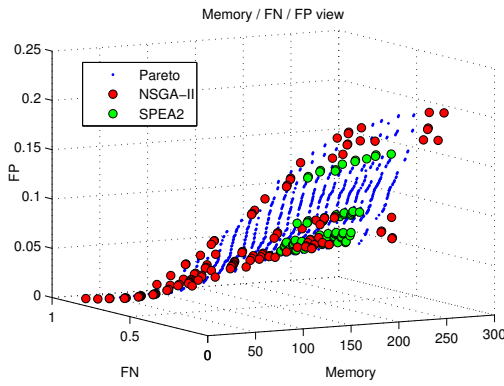


Figure 2: Solutions found by MOEAs for sparse topology using evolution parameters No. 45.

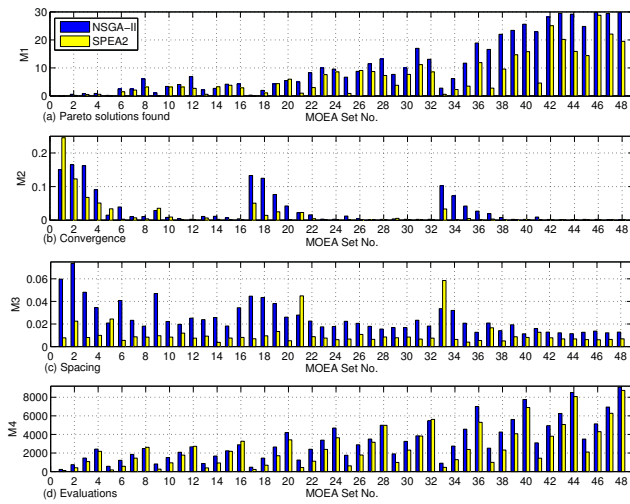


Figure 3: Results of metrics for sparse topology.

We found out that higher crossover probability has a higher impact on the speed of convergence than mutation probability from the perspective of metric  $M_2$ . See quartets in Fig. 3 (b). It is possible to obtain good results with high crossover probability and low mutation probability, but much more difficult if the crossover probability is very low. However, both parameters, as well as the population size, have a positive impact on the convergence.

**Diversification** There is a difference between *diversification* of NSGA-II and SPEA2 for our problem. Fig. 3 (c) suggests that the solutions are spread better using SPEA2. However, checking the objective space (Fig. 2) provides additional information on spreading of the solutions within the whole objective space that is better for NSGA-II. Note that SPEA2 found better spread solutions if mutation probability is higher.

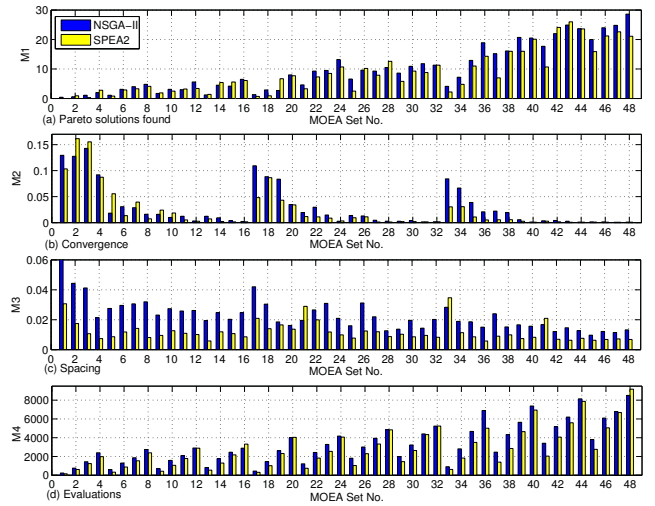


Figure 4: Results of metrics for dense topology.

**Evaluations** The number of simulations needed to evaluate the individuals in the population using a simulator are depicted in Fig. 3 (d). All MOEA parameters influence the number of evaluations, but  $PMut$  and  $PopSize$  have a higher impact than  $PCross$  (see, e.g., sets No. 36, 40, 44 and 48 with different values of  $PCross$  vs. sets 33 – 36 with different values of  $PMut$ ).

Note that, e.g., set No. 45 required only 3,497 simulations for NSGA-II and 2,101 simulations for SPEA2 in average. 24.8 and 14.4 Pareto dominant solutions were found for NSGA-II and SPEA2, respectively. Since the solutions cover different parts of the Pareto front (especially for NSGA-II), the evolution is much more efficient than evaluation of 25,000,000 configurations in case of exhaustive search.

### Dense Topology

In the *dense topology*, a node  $b_j \in C \cup A$  has 127 neighbours in average.

**Exhaustive Search** The true Pareto front obtained for the *sparse topology* can be found in (Stehlik et al., 2013). The results of the exhaustive search showed that 20,072 IDS configurations (out of 25,000,000) are Pareto dominant resulting in 2,219 unique solutions in the objective space.

**Convergence, diversification and evaluations** Results for evolution sets No. 45 and 48 can be found in (Stehlik et al., 2013). Similarly to the *sparse topology*, optimized solutions are spread better for SPEA2 in case of evolution set No. 48. The experiment showed similar characteristics of performance based on evolution settings as for *sparse topology*. The results of all performance metrics are shown in Fig. 4.

## Conclusion

In this work, we extended our *optimization framework* to utilize MOEAs in the process of IDS parametrization. The multi-objective approach is beneficial since it provides the network operator with a set of optimized solutions. He/she can select a solution according to the purpose of the WSN and change the IDS settings according to current needs. We showed that the knowledge of the Pareto optimal solutions is advantageous. However, the computation using exhaustive search is extremely time-demanding. MOEAs proved to be a good compromise between the quality of Pareto front approximation and the optimization time.

We compared two widely used MOEAs: NSGA-II and SPEA2. The results suggest that NSGA-II might be better for our needs. However, one should be careful with a definite conclusion. Various metrics, as well as visualization of the objective space, provide different views of the algorithm performance.

We also focused on the impact of MOEA parameters on the speed of convergence, number of evaluations and quality of Pareto front approximations. Higher population size provides better results at the cost of higher number of evaluations. We found out that a higher crossover probability does not increase the number of evaluations as much as a higher mutation probability and has a better impact on the quality of Pareto front approximations.

In the future, we plan to optimize techniques to detect other attacks than selective forwarding. We would also like to extend the optimization framework to design robust solutions in complex environments. Finally, we plan to adapt and use our framework for the optimization of a whole network stack.

## Acknowledgments

We would like to thank Lukas Sekanina for his invaluable advice. This work was supported by the GAP202/11/0422 project of the Czech Science Foundation.

## References

- Anderson, D. (2001). BOINC: a system for public-resource computing and storage. In *Proceedings of IEEE/ACM Workshop on Grid Computing*, pages 4–10.
- Crossbow (2013). MicaZ Datasheet. [Online; accessed 4/12/2013]. <http://bullseye.xbow.com:81/Products/productdetails.aspx?sid=164>.
- da Silva, A. P. R., Martins, M. H. T., Rocha, B. P. S., Loureiro, A. A. F., Ruiz, L. B., and Wong, H. C. (2005). Decentralized intrusion detection in wireless sensor networks. In *Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, pages 16–23.
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.
- Dressler, F. (2007). *Self-Organization in Sensor and Actor Networks*. John Wiley & Sons.
- INRIA Lille (2013). DOLPHIN project-team. ParadisEO - A Software for Metaheuristics. [Online; accessed 4/12/2013]. <http://paradiseo.gforge.inria.fr/>.
- Karlof, C. and Wagner, D. (2003). Secure routing in wireless sensor networks: Attacks and countermeasures. *AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, Elsevier, 1(2):293–315.
- Köpke, A., Swigulski, M., Wessel, K., Willkomm, D., Haneveld, P. T. K., Parker, T. E. V., Visser, O. W., Lichte, H. S., and Valentin, S. (2008). Simulating wireless and mobile networks in OMNeT++ the MiXiM vision. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems*, pages 1–8.
- Liefooghe, A., Jourdan, L., and Talbi, E.-G. (2011). A software framework based on a conceptual unified model for evolutionary multiobjective optimization: ParadisEO-MOEO. *European Journal of Operational Research*, 209(2):104–112.
- MOGALib (2013). MOGALib – MuleGA. [Online; accessed 4/12/2013]. <http://mulega.uni-pannon.hu/index.php/MOGALib>.
- Rappaport, T. (2001). *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 2nd edition.
- Roman, R., Zhou, J., and Lopez, J. (2006). Applying intrusion detection systems to wireless sensor networks. In *IEEE Consumer Communications & Networking Conference (CCNC 2006)*, pages 640–644, Las Vegas (USA).
- Schott, J. (1995). Fault tolerant design using single and multicriteria genetic algorithm optimization. Master’s thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics.
- Stehlik, M., Saleh, A., Stetsko, A., and Matyas, V. (2013). Multi-Objective Optimization of Intrusion Detection Systems for Wireless Sensor Networks – webpage. [Online; accessed 4/14/2013]. <http://www.fi.muni.cz/~xstehl2/ECAL/>.
- Stetsko, A. (2012). *On intrusion detection in wireless sensor networks*. PhD thesis, Masaryk University.
- Stetsko, A., Stehlik, M., and Matyas, V. (2011). Calibrating and comparing simulators for wireless sensor networks. In *Proceedings of the 8th IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 733–738, Los Alamitos, USA.
- Talbi, E. G. (2009). *Metaheuristics - From Design to Implementation*. Wiley.
- Zhang, Y. and Lee, W. (2000). Intrusion detection in wireless ad-hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking, MobiCom ’00*, pages 275–283, New York, NY, USA. ACM.
- Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: improving the strength pareto evolutionary algorithm. Technical report, Eidgenössische Technische Hochschule Zürich (ETH).

# Evolving Plastic Neuromodulated Networks for Behavior Emergence of Autonomous Virtual Characters

Yuri L. B. Nogueira<sup>1</sup>, Carlos Eduardo F. de Brito<sup>1</sup>, Creto A. Vidal<sup>1</sup> and Joaquim B. Cavalcante-Neto<sup>1</sup>

<sup>1</sup>Department of Computing, Federal University of Ceará, Brazil  
yuri@lia.ufc.br

## Abstract

This paper addresses the problem of generating natural behavior of autonomous virtual characters. Inspired by the fields of Embodied and Enactive Artificial Intelligence, we postulate that natural behavior is the result of a coupling between the agent and the world where it lives, which leads to a coherence between its actions and its surroundings. In this work, we present the tools that we have been using to study that idea: a controller based on a plastic neuromodulated neural network, which is capable of molding itself to received stimuli; and a simple novel method for genetic encoding of artificial neural networks. We show the capabilities of the controller in generating interesting foraging behavior of an autonomous virtual robot, and discuss the advantages of its emergent characteristics when compared with traditional approaches.

## Introduction

An important requirement for a user's sense of immersion in a virtual world is the way things move in the environment, so that it can be truly perceived as the real world. Zahorik and Jenison (Zahorik and Jenison (1998)) put it in this way: "When the environmental response is perceived as lawful, that is, commensurate with the response that would be made by the real-world environment in which our perceptual systems have evolved, then the action is said to successfully support our expectations".

The term "lawful" is appropriate to describe the behavior of inanimate elements that follow the physical laws. However, what is taken to be natural for the living elements, seems to be the opposite of simply following rules. The behavior of autonomous virtual characters plays an important role in virtual reality environments and obtaining such behavior in a natural and realistic way is still an open problem. We argue that unnatural behaviors are typically obtained by a lack of connection between the character and the world around it in rule-based implementations.

In this paper we propose:

- A controller that is capable of adapting itself to the character's bodily constitution and to the characteristics of the

environment, causing the emergence of appropriate foraging behavior in an autonomous virtual robot; and

- A simple novel method for genetic encoding of artificial neural networks (ANNs).

In the next section, we discuss the traditional methods used by the virtual reality community in generating behaviors of virtual characters, analyze the characteristics of those behaviors, and present some thoughts on the proposition that an emergentist approach could help to overcome the limitations resulting from approaches that attempt to explicitly model the cognitive abilities of the agents.

We also describe the tools that have been employed for studying the emergence of foraging behavior:

- A neuromodulated network for processing sensorial information and generating signals for motor action, which is capable of local modulation of synaptic plasticity; and
- A genetic algorithm (GA) evolution mechanism, which selects the most adapted neural networks.

We show our results with a simulated Khepera-like robot, equipped with sensors that input the robot's distance to a fruit or poison to the neural network. The robot's motor skills are moving forward and backward, and turning left or right. The experiments show that the robot was capable to learn how to make proper use of its sensors and motors in order to catch fruit and avoid poison, displaying a complex navigation control behavior. In conclusion, we present final discussions about the obtained results and future works on our study about natural behaviors of virtual characters.

## Behavior Generation of Virtual Characters

Our basic assumption is that the behavior of a virtual character can only be called realistic if it reflects the details of the agent's bodily constitution, and has a close and logical association with the events taking place in the virtual environment. Accordingly, this work supports the idea that the realistic behavior should be obtained as the result of a permanent and intimate dialogue between the agent and its en-

vironment. This position is motivated by the Embodied (Anderson (2003)) and Enactive (Froese and Ziemke (2009)) approaches to AI, where intelligence is perceived as something that emerges from a close interaction among all the components involved in the cognitive phenomenon.

Here, we have a clear contrast with the classical strategies for the simulation of behavior, which concentrate their efforts on embedding the relevant aspects of reality within the agent's mind (in the form of models and "knowledge"), and internally calculating what appropriate action should be performed in a given situation (Shao and Terzopoulos (2007); García-Rojas et al. (2007); Gutierrez et al. (2007); Whiting et al. (2010); Orozco et al. (2011)). The main drawback of that approach is the great difficulty of maintaining a complete and updated description of the relevant aspects of reality, especially in the case of highly dynamical environments. So, a typical result is the strong attachment of the agent's behavior to the rules and facts stored in its mind, rather than an involvement and interaction with what is really happening in its surroundings. That generates a feeling of detachment, which we sometimes classify as robotic behavior.

The diagram in Figure 1a illustrates the traditional approach for behavior generation. That approach subdivides the problem into a cognitive level and a motor level, modeling each of those parts separately. In that view, the cognitive level works as a calculator of abstract facts, i.e., the manipulation of symbols already interpreted by the programmer, which correspond to a high-level perception of the world. That calculation should provide the correct (or desired) command to the motor level, which has a built-in set of behaviors fully described, such as walking, sitting or standing in line. The so called reactive agents are typically implemented based on this two-level modelling. To mitigate the problems inherent with this approach, some works (Pina et al. (2006); Schneider and Rosa (2009)) attempt to use techniques of machine learning to automate the process of behavior selection.

To avoid the detachment problem, we propose that the behavior of autonomous virtual characters should be generated through a methodology of emergence. More specifically, we suggest that efforts should migrate from detailed internal representations of reality and explicit descriptions of behavior, to the construction of appropriate conditions that would induce the coupling of the structure and dynamics of mind, body and environment, resulting in the emergence of intelligent behavior. In this way, we expect to obtain the degree of coherence between agent and environment which is required for natural behavior.

The notion of an emergent property of a system refers to a global characteristic of the system that cannot be found in any of its parts (Klaus and Mainzer (2009)). Accordingly, by emergent behavior we mean that the characteristics of the behavior that emerges are not described or encoded in any of the components that define the system: the logic of the controller, the anatomy of the body, or the environment con-

figuration. In fact, the behavior must result from the combination of the properties of all those components.

The works based on the emergentist approach (Sims (1994); Chaumont et al. (2007); Nogueira et al. (2008); Panzoli et al. (2010); Palmer and Chou (2012)) use two basic elements to support the emergence of high-level behaviors: the uninterpreted signals received from the environment, and the elementary movements performed by the body parts. The internal dynamics of the body and mind of the agent should then modulate this information in order to establish a sensorimotor flow that coordinates the low-level movements in order to produce the emergence of high-level behavior. This idea is illustrated in Figure 1b. The key aspect that makes this process possible is that some components have a degree of plasticity, so that their structures and internal dynamics can be modified over time, in response to interactions with other components of the system, in an evolutionary dynamics.

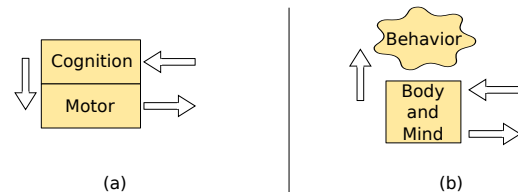


Figure 1: Approaches to behavior generation. The two-level modeling (a), and the emergentist approach (b).

Emergence phenomena tend to produce very specific configurations which are sensitive to small changes in the elements that constitute the system. This observation is consistent with the fact that animals of different species sometimes solve the same problem in radically different ways. This is illustrated by the great diversity of behaviors of locomotion, hunting, breeding, etc. that we find in nature. If the assumption that emergent phenomena are an important aspect of behavior, that would explain, in part, why the problem of realistic simulation has resisted for so long to approaches based on traditional AI techniques: the details of the situation are what determine the behavior of the agent. But these details are precisely what is left out in the process of abstraction inherent to representationalism. In other words, the attempt to simulate the behavior of an agent based solely on a high-level description of the situation in terms of goals and motivations of the agent, complex actions that the body can perform, and qualities of objects and features of the environment, leads almost inevitably to mechanical and unnatural behavior.

## The Controller

Here, we present a controller for generating behaviors of virtual characters based on emergence. As we argued, to achieve the emergence of behavior, an essential factor is the



generation of a dynamics which is capable of modifying and adapting the controller to the body of the character and to the environment around it. Aiming at this goal, we chose to evolve a plastic neuromodulated neural network to generate the signals that control the motors of a virtual robot based on its sensory information. The neuromodulatory property brings lifetime adaptation to the network, while a genetic algorithm evolves the network providing adaptation throughout the generations.

## The Neural Network

The neural network that we used is essentially a Continuous Time Recurrent Neural Network (CTRNN), whose neurons are modeled in the following general form (Beer (1995)):

$$\frac{dy_i}{dt} = \frac{1}{\tau_i}(-y_i + \sum_{j=1}^n w_{ji}f(s_j) + I_i) \quad (1)$$

where  $t$  is time,  $y_i$  and  $\tau_i$  are, respectively, the internal state and time constant for each neuron  $i$ ,  $w_{ji}$  is the weight of the  $j$ th input synapse of neuron  $i$ ,  $s_j$  is the state of the neuron linked to the  $j$ th input synapse,  $f()$  is the activation function of a neuron and  $I_i$  represents a constant external input to neuron  $i$ . In this work we used the same  $I$  for all neurons, simulating an external stimulus from a higher center such as the mesencephalic locomotor region of animal brains.

In addition to this standard dynamics, we chose to include a neuromodulatory feature in the neural network. As discussed by Soltoggio et al. (Soltoggio et al. (2007)), the neuromodulation plays an important role in neural substrates from invertebrates to the human brain, and is related to the induction of Late phase - Long Term Potentiation (L-LTP), a phenomenon of permanent growth of the synaptic contact in brain, causing synaptic stability, being a potential candidate for explaining memory functions involving neural wiring and, consequently, learning. Another mechanism of neural wiring related to neuromodulation is the Long Term Depression (LTD), the permanent decrease of the synaptic contact (Soltoggio et al. (2008)).

Soltoggio et al. proposed the use of a model of neuromodulation in T-Maze (Soltoggio et al. (2008)) and in bee foraging behavior (Soltoggio et al. (2007)). We use the same model in our work, in order to study the effects and advantages of such dynamics in the behaviors of our virtual characters. It consists of a CTRNN with two types of neurons: standard neurons, which are the processing units, and modulatory neurons, which are responsible for modulating the changes of weights in the synapses following the Hebbian rule. Figure 2 shows the modulation mechanism.

Equation 1 defines the activities of the standard and modulatory neurons. The signals of both types of neurons are computed according to that equation, with the activation function  $f(x) = \tanh(x/2)$ . However, only the input signals that come from standard neurons, and the weights of

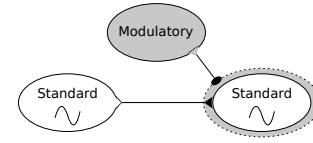


Figure 2: The network processing itself is made only with the standard neurons. The modulatory neuron determines the response of the Hebbian rule, mediating the amount of growth or decreasing of the synaptic weight.

the respective synapses, are considered for the summation in that equation.

The value of modulatory activation  $m$  acting on a neuron  $i$  is computed as follows:

$$m_i = \sum_{j \in Mod} w_{ji} \cdot \tanh(s_j/2) \quad (2)$$

where  $j$  represents all modulatory neurons connected to neuron  $i$ ,  $w_{ji}$  is the weight associated with the synapse that links modulatory neuron  $j$  to neuron  $i$  and  $s_j$  is the internal state of modulatory neuron  $j$ .

Finally, the synapse's weight change is defined by the following equation:

$$\Delta w_{ji} = \tanh(m_i/2) \cdot \eta \cdot [A o_j o_i + B o_j + C o_i + D] \quad (3)$$

where  $\Delta w_{ji}$  is the amount of change in the synapse that links the standard neuron  $j$  to a neuron  $i$ ,  $\eta$  is the learning rate,  $A$ ,  $B$ ,  $C$  and  $D$  are tunable parameters, and  $o_j$  and  $o_i$  are the activation function  $\tanh(x/2)$  applied to internal states of neurons  $j$  and  $i$  respectively. Note that this rule is unstable if the parameters are such that  $\Delta w_{ji}$  is always positive or always negative. To avoid that, we limit the weights of all input synapses of each neuron. Equation 3 differs from the Hebbian rule only on the modulatory term, i.e.,  $\tanh(m_i/2)$ . That is, the modulatory effect changes the learning rate of the Hebbian rule.

The model presented here differs from that of Soltoggio's work in the way we define the input and output neurons. Our work defines some standard neurons as afferent neurons or efferent neurons, which have no internal dynamics. The internal state of an afferent neuron is a sensor value and cannot receive input from other neurons, while an efferent neuron stores the average of the internal states of each neuron connected to it, defining the outputs of the network.

## The Evolutionary Algorithm

**Evolving neural networks** In the literature, we find three main general techniques to evolve neural networks with GA aiming at topology search (Mattiussi and Floreano (2007)): direct encoding, developmental encoding and implicit interaction. In this work, we propose a novel simple method based on the third technique to search appropriate values

to the weights, the time and tuning parameters, the learning rate, the neurons to define as input or output, and the topology of our ANN.

Direct encoding is a straightforward approach, in which the chromosomes describe exactly the graph of the neural network. That requires complex gene and genetic operators specifications oriented to graph evolution, aiming at the maintenance of its consistency, which makes it difficult to simultaneously evolve several characteristics not naturally describable with this type of structure. A well-known algorithm based on direct encoding is the NeuroEvolution of Augmenting Topologies (NEAT) (Stanley and Miikkulainen (2002)).

The developmental encoding genome describes the rules of a developmental process of the network, i.e., the process of constructing and growing the network. This approach resembles genetic programming and allows a compact description of the network. However, it is very difficult to define the genetic operators, and a poor definition can lead to unsatisfactory results in terms of evolution.

Implicit interaction is a biologically inspired technique based on the fact that the interaction between genes in the DNA is the key to the definition of their expressions. The Analog Genetic Encoding (AGE) (Mattiussi and Floreano (2007)) is a well-established encoding based on this idea. The genome is represented as a simple string of characters, allowing the use of traditional genetic operators such as crossover and mutation, while keeping the structure and consistency of the network. Since the encoding we propose is based on AGE, next we describe it in more detail.

In AGE, while neurons are explicitly described in the chromosome, synapses are implicitly defined, since they are formed by the interaction between genes, and not by a gene itself. The genes specify the neurons and their respective terminals, i.e., their inputs and outputs, while interactions between terminals form the synapses. To assemble the synapses, the inputs of all neurons are aligned with the outputs of all neurons and an alignment score is calculated, which indicates the weight of the synapses between the aligned neurons.

Each element of the neural network is encoded in the AGE's chromosomes into substrings called tokens. For example, to encode a neuromodulated network, Soltoggio et al. (Soltoggio et al. (2007)) used the tokens NE to indicate a standard neuron, MO to indicate a modulatory neuron and TE to delimit a terminal sequence.

A terminal sequence is an arbitrary sequence of characters that precede the token TE and defines a neuron's terminal. After a neuron token, all the subsequent characters until a TE token are translated as a neuron's terminal. Each appearance of TE determines a neuron's terminal, and, so, the neuron has as many terminals as the number of TE's appearances after it.

AGE is supposed to accomplish the evolution of any type

of analog network, such as electronic networks, neural networks and genetic regulatory networks. To do that, the alignment score is based on a network-specific interaction map that leads to a complex chromosomal representation. Our proposal focuses on ANNs and specifies the chromosome as a binary array, encoding the parameters of the network in a more straightforward way, using a simpler similarity function, and still maintaining the advantageous properties of AGE's interaction maps (Mattiussi and Floreano (2007)) for ANNs evolution. Such an encoding scheme allows us to easily search augmenting topologies of neural networks composed of different types of devices (neurons).

**The Proposed Genetic Encoding** Our chromosomes are arrays of bits, with each group of 32 bits defining a gene. The first 8 bits (1 byte) of the gene are used to encode an identifier, which indicates the element that the gene represents in the network. The last 24 bits specify a value that indicates a property of the decoded element.

Each individual in the population has two chromosomes: one chromosome stores the global parameters of the neural network, while the other keeps the network itself. The global parameters are the values of the variables in Equation 3 and the external stimulus of the CTRNN (Equation 1). Figure 3 shows the distribution of values in the chromosome.

I	$\eta$	A	B	C	D
---	--------	---	---	---	---

Figure 3: Encoding of the global parameters of the network.

To decode the network chromosome, we have to read each gene and isolate its identifier and value. The 8 bits of the identifier are decoded according to Table 1. The 24 bits that encode the value of a gene are linearly mapped into a floating-point value  $v$  in the range  $[-1, 1]$ , according to the formula:

$$v = \left( \frac{n}{2^{24} - 1} \cdot 2.0 \right) - 1.0 \quad (4)$$

where  $n$  is the unsigned integer encoded in the value bits.

Table 1: Genes' identifiers	
Byte	Meaning
$0 \leq id \leq 38$	Standard Neuron (SN)
$39 \leq id \leq 51$	Modulatory Neuron (MN)
$52 \leq id \leq 255$	Neuronic Terminal (TR)

With the distribution of identifiers shown in Table 1, we have the following probabilities:  $P(SN) = 0.15$ ,  $P(MN) = 0.05$  and  $P(TR) = 0.80$ , assuming a randomly generated chromosome. This distribution was chosen because we want to have more standard neurons, which actually do the signal generation, than modulatory neurons, which only change the plasticity. At the same time, we need

to have more synapses than neurons, and, for this reason, the probability of creating terminals is greater than that of creating a neuron. The probabilities were empirically chosen.

Suppose that the robot we want to control has  $s$  sensors and  $m$  motor outputs. To keep some structure of the neural network in the chromosome, we fix the first  $s$  genes that encode SN to be afferent neurons, while we set the last  $m$  genes that encode SN to be efferent neurons.

The value of a gene (i.e., the last 24 bits) identified as a SN or a MN, encode the time constant  $\tau$  in Equation 1. The exceptions are the afferent neuron, whose value identifies a stimulus parameter, and the efferent neuron, whose value is ignored, since it has no internal dynamics. The stimulus parameter is  $-1$ , if  $v < 0$  or  $1$ , if  $v \geq 0$ , and is multiplied by the sensor signal, generating excitatory or inhibitory sensory signal as input to the neural network.

The genes identified as TR have the value part corresponding to the input or output terminals of the last-read neuron. The first TR read after a neuron's gene is always its input, while the second TR is always its output. Such genes are ignored if no neuron was read before them or the last-read neuron already has its terminals. However, their presence in the chromosome is useful to generate new combinations by the genetic process.

The parsing of a chromosome produces a list of neurons with their respective parameters and terminals that will compose the network. Similar to what is done in AGE, we pair neurons by their terminals to create the synapses arriving at or leaving the neurons. However, instead of using an interaction map, we define the synapses' weights according to a similarity measure. The idea is to use a proximity function, computing a distance value of two numbers based on the Hamming distance between their binary representations, according to equation:

$$w(i, o) = \frac{eb \cdot (i + o)}{nb \cdot 2}, \quad (5)$$

where  $w$  is the weight of a synapse that links an output of value  $o$  with an input of value  $i$ . The symbol  $nb$  indicates the total number of bits that represent the value (24 bits), and  $eb$  is the number of equal bits at the same position between the binary representations of  $i$  and  $o$ . We also defined an existence condition empirically to increase topological diversity: if  $\lfloor eb/4 \rfloor \bmod 3 = 0$  then  $w(i, o) = 0$ .

Equation 5 may be interpreted as: the synapse weight is the average of the input and output parameters weighted according to distance. The occurrence of two terminal parameters with zero equal bits, implies a maximum distance from each other, and, therefore, there is no synapse linking them. The idea of the similarity function is that if the number of equal bits is the maximum possible, then  $nb = eb$ , which implies that the synapse weight is the average of both parameters. Equal parameter values imply a synapse weight with the same value.

In order to understand the decoding process better, take Table 2 as example of an excerpt from a chromosome. Analyzing the identifiers based on Table 1, we have two neurons in the chromosome: an SN on gene G1 with value 15,000,000, and an MN on gene G4 with value 8,500,000. Using Equation 4 with those values, we obtain  $\tau_{SN} = 0.79$  and  $\tau_{MN} = 0.01$ .

Table 2: Sample excerpt from a Network Chromosome

	G1	G2	G3	G4	G5	G6
Identifier	30	120	240	40	125	200
Value	15M	10M	4M	8.5M	11.5M	6M

Genes G2 and G3 are the input and output terminals of SN, while G5 and G6 are the terminals of MN in the same order. From Equation 4, we have the values  $TR_{G2} = 0.19$ ,  $TR_{G3} = -0.52$ ,  $TR_{G5} = 0.37$  and  $TR_{G6} = -0.29$ . Figure 4 illustrates the resulting neural organization.

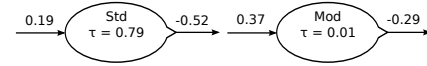


Figure 4: Neurons decoded from Table 2.

Finally, the synapses' weights of the network are computed with Equation 5 applied to each pair input-output, i.e.:  $w(TR_{G2}, TR_{G3})$ ;  $w(TR_{G2}, TR_{G6})$ ;  $w(TR_{G5}, TR_{G3})$ ;  $w(TR_{G5}, TR_{G6})$ .

Let us compute  $w(TR_{G2}, TR_{G3})$  as an example. The binary representation of those parameters, i.e., the way they are stored in the chromosome, are:

$$\begin{aligned} \text{bin}(0.19) &= \text{bin}(10M) = 100110001001011010000000 \\ \text{bin}(-0.52) &= \text{bin}(4M) = 0011111010000100100000000 \end{aligned}$$

From these representations, we have  $eb = 13$ . Thus, using Equation 5, we would have  $w(0.19, -0.52) = -0.09$ . However, the existence condition,  $\lfloor 13/4 \rfloor = 3$  implies that  $3 \bmod 3 = 0$ , and thus there is no synapse from standard neuron to itself. Similarly computing the other weights, we will have only one synapse, from MN to SN, with  $w(TR_{G2}, TR_{G6}) = -0.03$ . Figure 5 shows the fragment of the network decoded from Table 2.

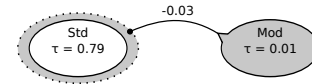


Figure 5: Network decoded from Table 2.

**The Simulation of Evolution** To evolve the individuals, we simply apply the canonical genetic algorithm. The relative chromosomes of the individuals are paired and the duplication, crossover and mutation operators are applied. Note that, since a chromosome is simply an array of bits,

crossover can break a gene, causing a new gene to appear. The simple mutation of a single bit also can lead to the appearing of a new gene.

Each individual is decoded and made alive for controlling a virtual robot. The robot has an amount of energy that is reduced proportionally to the strength of the generated signals. The evaluation function we used for the emergence of foraging behavior was the number of collected food multiplied by the robot's lifespan.

## Case Study

### Description of Experiment

To evaluate the proposed controller, we put it to control a Khepera-like virtual robot in a foraging task. The environment consists of randomly distributed fruits and poisons. The simulation was developed with the *Irrlicht 3D Engine*<sup>1</sup>, with physics provided by the *Bullet Physics Engine*<sup>2</sup>.

The robot is shown in Figure 6. It has a cylindrical body with a black box that plays the roles of eye and mouth. The robot "eats" a fruit or a poison if that box touches them.

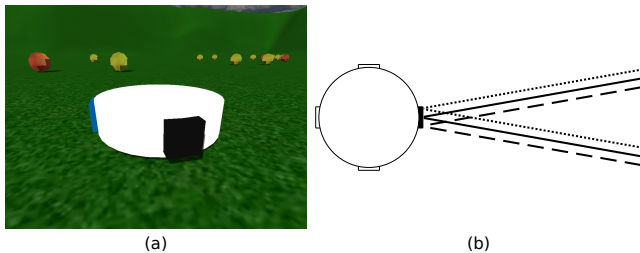


Figure 6: The robot. (a) Robot in the environment. The black box is its eye and mouth. (b) The three sensors distributed in the eye and their fields of sense. We used 20° Field Of Sense (FOS). The maximum sensing distance is six times the robot's diameter.

The neural network is connected with the body and the environment receiving as input the signals of the robot's sensors. There are three sensors aligned side by side at the extent of the eye (Figure 6b), each one able to catch the normalized distance  $([0, 1])$  to the nearest fruit and poison inside its FOS. This implies the generation of six values. There is also a proprioceptive sense of energy, that enables the robot to sense its level of energy spending, which ranges from 0 (the robot is fully energized) to 1 (the robot is totally exhausted). In this way, the strength of the signal allows the robot to perceive when its energy is finishing. This ANN then needs seven afferent neurons.

The robot has two motors, one to move forward or backward and another to make left or right turns, each one controlled by one efferent neuron. When the first motor receives a positive value, the robot moves forward and if it receives

a negative value, the robot moves backward. With a positive value, the second motor makes the robot turn right, otherwise it causes the robot to turn left. The robot's energy is reduced every simulation step according to Equation 6, where  $o_1$  and  $o_2$  are the output of the two efferent neurons of the ANN. The constant value 10 is used in order to avoid the evolution of stationary robots.

$$C = (|100 * o_1| + |100 * o_2|)^2 + 10. \quad (6)$$

A robot starts with 50,000 energy units (eu). This value increases 10,000eu whenever a good fruit is eaten (up to a maximum value of 50,000eu) and decreases in two situations: (1) when the robot is alive, its energy is continuously decreasing in proportion to the applied motor signals, (2) whenever the robot eats a poisonous fruit, its energy is reduced to 10,000eu. In the second situation, if the robot's energy level is less than or equal to 10,000eu, the energy is zeroed. If the energy is exhausted, the robot dies.

Each controller decoded from the chromosomes of an individual is assigned the control of a robot, one at a time. Each trial begins at the same position, and the fruits and poisons are always randomly redistributed to prevent the GA from "memorizing" the positions. A trial ends when the robot's energy is exhausted. The GA randomly generates the first population. The following parameters were used:

- Population size: 100 individuals
- Network chromosome size: 100 genes (3200 bits)
- Type of crossover: Monopoint (one break point)
- Crossover probability: 60%
- Mutation probability (per bit): 0.1%

## Results

Due to space constraints, we will focus on the analysis of type and quality of the generated behaviors and we will not show the measurement data and comparisons of the several runs we made. However, in all executions we made, the same behavioral result was obtained, except for the neuro-modulatory action, that have been developed in rare cases, as we will discuss later. The plots shown in this section represent the typical results of our experiments.

The GA successfully evolved the neural network to control the robot in the foraging task. Figure 7 shows the evolution of the evaluation averages of all the individuals per generation.

The robot successfully acquired the behavior of catching good fruits only, while avoiding the poisonous fruits, as shown in Figure 8a. Note that, although the evaluation function of the genetic algorithm explicitly selects those individuals that collect the greater number of fruits, there is no direct information about poisons. However, we can observe

<sup>1</sup><http://irrlicht.sourceforge.net/>

<sup>2</sup><http://bulletphysics.org/>



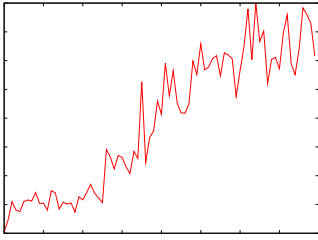


Figure 7: Evolution of the evaluation averages of the populations.

the behavior of moving away from the poisonous fruits (Figure 8b), as a consequence of the fact that individuals who eat poisons may “die” sooner.

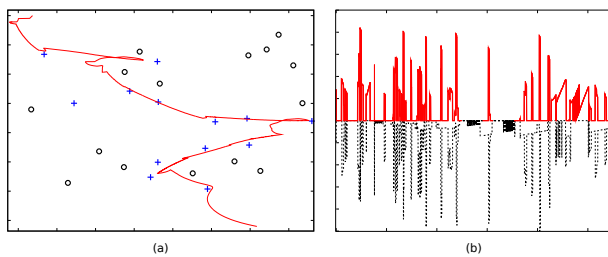


Figure 8: Robot Behavior. (a) The line represents the robot's path, starting at (0,0). Note that its path passes through the fruits (+) while deviates from poisons (o). The line is not always touching the fruits because the graph is showing the center of the objects and, in the simulation, the eye's mesh only needs to touch the fruit's mesh. (b) Motor and poison sensor activities. Note that when a poison is sensed (positive values of solid line), there is a peak of negative signal on motor activity (dotted line), which leads to a backward movement.

An important characteristic to point out about the robot's behavior is how the high level foraging behavior is performed with low level behaviors of direction adjustment. One sensor of the eye alone cannot determine the direction to follow in order to catch the sensed fruit, since it only captures the distance to the fruit. Therefore, the robot needs to change its position to be able to use the three sensors to find out the missing information. This behavior is shown in Figure 9, where we can see the robot turning left to use the right side of a sensor's FOS to follow the fruit. Since the three sensors are slightly displaced with respect to one another, when a fruit leaves the FOS of a sensor, it is possible to determine to what side an adjustment of direction needs to be made. (Figure 6b).

Regarding the modulatory activity, one particular controller evolved with modulatory neurons. With this action, the robot exhibited two ways of search. A local rotation, searching for near food and, when no fruit was caught, it

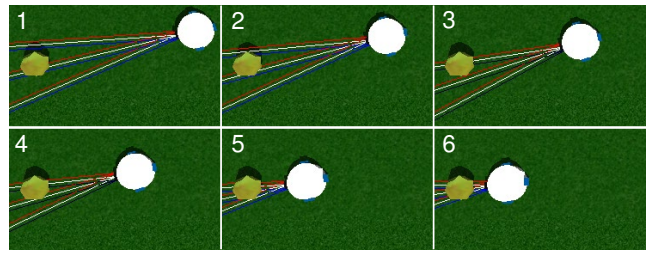


Figure 9: Direction adjustment behavior. Note that, to catch the fruit, the robot approaches it using the right side of the FOS, instead of the FOS's center. The top frames show the moment that it senses the fruit and then turns left. The bottom frames show the robot catching the fruit following its “side sensing”.

gradually increased its rotation radius, until it found a fruit, and then passed to the local search again. Figure 10 shows the joint activity of the modulatory neuron, stimulated with the energy sense (signal continuously increasing while no food is collected), and the changing of the motor activity.

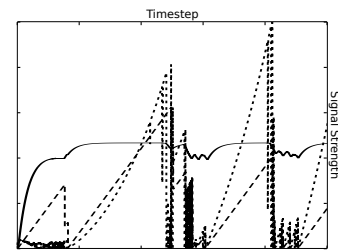


Figure 10: Motor activity (dotted line), energy sense signal (dashed line) and modulation activity (solid line). Note that these three activities are synchronized. When the robot catches a fruit (decrease in dashed line), it searches another one locally (decrease in dotted line). If no fruit is eaten, it gradually increases the search radius (increase in solid and dotted lines).

## Conclusion

We described a controller for behavior generation of autonomous virtual characters. We argue that natural behaviors can emerge if the behavior controllers are designed properly, taking into account emergence principles. Such a controller must be capable of adapting itself to the body and to the environment. Thus, it needs to be able of modify itself in contact with the world.

The controller uses neuromodulation for changing its dynamics on the fly, while adapted throughout the generations by the genetic algorithm. In animal brains, the neuromodulators are directly related to memory functions and indirectly to learning. In our experiment, they allowed modifications in behaviors patterns according to environmental changes.

We also presented a simple novel way of genetic encoding ANNs, describing simple arrays evolvable with a canonical genetic algorithm. Those arrays are able to evolve neural networks with growing topologies, and, at the same time, is possible to evolve multiple characteristics of an agent.

We showed the capabilities of our controller through an application involving the foraging behavior of a virtual robot. The robot was able to learn how to use its own movements to compensate for the insufficiency of sensory data in order to accomplish the objective of catching fruits, showing a complex foraging behavior consisting of minor position corrections toward the goal. It is worthwhile to point out that when the plasticity of the controller was increased with modulatory actions, more elaborate strategies have emerged.

The results of our experiments show that the emergentist approach is indeed capable of producing the intimate coupling between agent and environment required for natural behavior. This fact is clearly illustrated by the strategy developed by the virtual agent to compensate for its primitive visual sensory apparatus, showing a high level behavior composed of minimal movements extremely connected with the conditions of the world, rather than a simple and straight “follow the fruit” behavior.

On the other hand, the simulations also show that the type of behaviors which we were able to obtain are relatively simple and, at this point of the investigation, it is not clear how to incrementally increase the complexity of the system in an emergent way. However, the traditional techniques can produce behaviors with arbitrary complexity, by using more detailed models and facts about reality, but paying the price of some level of detachment with respect to the environment. So, a natural question is whether it is possible to combine ideas of the traditional and emergent approaches to obtain the advantages of both sides.

## Acknowledgements

This work was supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

## References

- Anderson, M. L. (2003). Embodied cognition: A field guide. *Journal of Artificial Intelligence*, 149:91–130.
- Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3(4):469–509.
- Chaumont, N., Egli, R., and Adami, C. (2007). Evolving virtual creatures and catapults. *Artificial Life*, 13:139–157.
- Froese, T. and Ziemke, T. (2009). Enactive artificial intelligence: Investigating the systemic organization of life and mind. *Journal of Artificial Intelligence*, 173:466–500.
- García-Rojas, A., Vexo, F., and Thalmann, D. (2007). Semantic representation of individualized reaction movements for virtual humans. *IJVR*, 6(1):25–33.
- Gutierrez, D., Frischer, B., Cerezo, E., and Serón, F. (2007). AI and virtual crowds: Populating the colosseum. *Journal of Cultural Heritage*, 8(2):176–185.
- Klaus and Mainzer (2009). From embodied mind to embodied robotics: Humanities and system theoretical aspects. *Journal of Physiology-Paris*, 103(3-5):296 – 304.
- Mattiussi, C. and Floreano, D. (2007). Analog Genetic Encoding for the Evolution of Circuits and Networks. *IEEE Trans. Evol. Comput.*, 11(5):596–607.
- Nogueira, Y. L. B., Vidal, C. A., and Cavalcante-Neto, J. B. (2008). A nervous system model for direct dynamics animation control based on evolutionary computation. In *SAC '08: Proc. of the 2008 ACM symposium on Applied computing*, pages 1793–1800. ACM.
- Orozco, H., Ramos, F., Ramos, M., and Thalmann, D. (2011). An action selection process to simulate the human behavior in virtual humans with real personality. *The Visual Computer*, 27:275–285.
- Palmer, M. E. and Chou, A. K. (2012). An artificial visual cortex drives behavioral evolution in co-evolved predator and prey robots. In *Proc. of the 14th intl. conf. on Genetic and evolutionary computation conference companion*, GECCO Companion '12, pages 361–364. ACM.
- Panzoli, D., de Freitas, S., Duthen, Y., and Luga, H. (2010). The cortexionist architecture: behavioural intelligence of artificial creatures. *Vis. Comput.*, 26:353–366.
- Pina, A., Serón, F., Cerezo, E., and Gutierrez, D. (2006). ALVW: an alife behaviour modelling system. *Kybernetes*, 35(9):1431–1451.
- Schneider, M. O. and Rosa, J. L. G. (2009). Application and development of biologically plausible neural networks in a multi-agent artificial life system. *Neural Comput. Appl.*, 18(1):65–75.
- Shao, W. and Terzopoulos, D. (2007). Autonomous pedestrians. *Graph. Models*, 69(5-6):246–274.
- Sims, K. (1994). Evolving 3D morphology and behavior by competition. *Artificial Life*, 1(4):353–372.
- Soltoggio, A., Bullinaria, J., Mattiussi, C., Dürr, P., and Floreano, D. (2008). Evolutionary advantages of neuromodulated plasticity in dynamic, reward-based scenarios. In *Artificial Life XI: Proc. of the 11th Intl. Conf. on the Simulation and Synthesis of Living Systems*, pages 569–576. MIT Press.
- Soltoggio, A., Dürr, P., Mattiussi, C., and Floreano, D. (2007). Evolving neuromodulatory topologies for reinforcement learning-like problems. In *In: Proc. of the IEEE Congress on Evolutionary Computation, CEC*.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.
- Whiting, J. S., Dinerstein, J., Egbert, P. K., and Ventura, D. (2010). Cognitive and behavioral model ensembles for autonomous virtual characters. *Comput. Intell.*, 26(2):142–159.
- Zahorik, P. and Jenison, R. L. (1998). Presence as being-in-the-world. *Presence-Teleop. Virtual Env.*, 7:78–89.

# GOLEM: Generator Of Life Embedded into MMOs

Andrea Guarneri<sup>1</sup>, Dario Maggiorini<sup>1</sup>, Laura A. Ripamonti<sup>1</sup> and Marco Trubian<sup>1</sup>

<sup>1</sup>Dept. of Computer Science – Università di Milano - Italy  
ripamonti@di.unimi.it

## Abstract

Massively Multi-player Online Role-Playing Games and Massively Multiplayer Online games are complex and costly system from both a game design and a technical point of view. Among their many issues, in this paper we tackle the problem of incrementing players thrill by supplying them plenty of freshly produced, unpredictable monsters. To achieve this goal we have designed, developed and tested GOLEM, a genetic-based approach to the evolution of new species of monsters for video games.

## 1. Introduction

In Massively Multi-player Online Role-Playing Games (MMORPGs) and in Massively Multiplayer Online games (MMOs) players interact among them in an online, persistent, and shared virtual world. This game genre has gained success and diffusion especially thanks to the prosperity of Blizzard's World of Warcraft (WoW) (Taylor, 2006). The huge amount of people that interact on an ongoing basis (e.g., more than 10 million in WoW) in a shared environment raises questions for game designers and computer scientists that have scarcely been investigated until now, but that could nonetheless affect the success and survival of a specific game.

Beside the issues related to the game mechanics and the service supply, there is a number of other – only apparently – minor, “tricky” features that could, nonetheless, impact deeply on players' satisfaction with the game (see e.g., Bartle, 2003; Maggiorini et al., 2012a, 2012b). Among these neglected issues, we can enlist several characteristics that are intrinsic to the paths players have to follow to raise the “level” of their characters and that could rise problems of players' loyalty. In particular, to explore the in-game world, complete quests, and gain “experience points”, players are often confronted with different types of monsters (also called “mobs”, which stands for “mobiles”), which they are supposed to slay. In spite of the fact that a game world can contain hundreds of different species of monsters, after spending a certain amount of time playing, players become well aware of the characteristics presented by each specie and its related hazard. In the long run, this knowledge has the drawback of generating a certain amount of boredom in players, which lose the thrill of braving unfamiliar dangers (Koster, 2004).

In this work, we tackle this issue by proposing GOLEM – Generator Of Life Embedded into MMOs: an algorithm aimed at increasing variety and unpredictability among mobs in MMOs, by means of an approach rooted into Genetic Algorithms (GAs). The main idea is to represent each monster

specie in the population present in the in-game world through its genome, and to generate new species by recombining their chromosomes in a way quite similar to what happens in the natural world. This implies also taking into account aspects like the actual possibility for a mob to survive the habitat in which it was born (e.g., a marine-like animal with fins will unlikely survive in a desert), providing an estimate of the population growth (in order to avoid overpopulation), some means to contain the mobs numerosity when needed, etc. Nonetheless, our aim is not to recreate a complex ecosystem, since this will go far beyond the scopes of a generator of monsters for a video game, whose main goal should simply be to increase the players' fun.

The paper is organized as follows: the following Section 2 briefly analyses related works, from both the academy and the industry, while Section 3 recalls the fundamental concepts related to Genetic Algorithms (GAs). The subsequent Section 4 summarizes how we have designed the chromosomes to describe the most diffused fantasy monsters. Section 5 concentrates on the issues related to managing a population of evolving monsters through several generations. Sections 6 and 7 focus, respectively, on the implementation of the GOLEM algorithm and on some perspective results derived from several tests. Finally, Section 8 draws conclusions and delineates major future developments.

## 2. Related works (in video games)

Although scholarly literature on the applications of GAs and Genetic Programming (GP) in video games is quite huge and multi-faceted, in our knowledge, until now it has focused on scopes different from creating diversity among mobs. In particular, the major part of recent works has tackled either the use of GAs to generate or evolve the environment (i.e. the in-game world or game levels), or the evolution of agents (the so-called “bots”) behavior in order to produce more challenging opponents to the players.

In the first category we can enlist, e.g., the work of (Frade et al., 2012), which develops a GP-based procedural content technique to generate procedural terrains that do not require parameterization. Taking a different perspective, (Halim & Raif Baig, 2011) propose a set of metrics for measuring entertainment in video games and then use evolutionary algorithms to generate games using the proposed entertainment metrics as the fitness function. (Mourato et al., 2011) propose an approach rooted into GAs to generate automatically levels for a video game. The only work



suggesting the use of GAs in MMORPGs – to the best of our knowledge – is (de Carvalho et al., 2010), which proposes to use GAs for managing the dynamics of geophysics events, asserting that specific events will occur only in areas where they are prone to. The work of (Frade et al., 2010) uses GP to evolve automatically Terrain Programs, which are able to generate terrains procedurally, for a set of desired accessibility parameters. Finally yet importantly (Sorenson & Pasquier, 2010) developed a genetic encoding technique specific to level design used to generate game levels.

In the second category, we enlist works focused on evolving bots' behavior. For example, (Mora et al., 2012) focus on an evolutionary algorithm designed for evolving the decision engine of a program that plays Planet Wars, a game requiring the bot to deal with multiple target, while achieving a certain degree of adaptability in order to defeat different opponents in different scenarios. (Esparcia-Alcázar & Jaroslav, 2012) focus their attention on the problem of estimating the fitness value of individuals in an evolutionary algorithm while containing time and costs. In the field of racing video games, (Onieva et al., 2012) developed a driving system, whose controller for adapting the speed and direction of the vehicle to the track's shape is optimized by means of a GA. The work of (Inführ & Raidl, 2012) showed how GP could be used to create game playing strategies for 2-AntWars, a deterministic turn-based two-player game with local information. (Barros et al., 2011) used GAs to develop a convincing artificial opponent for chess. (Benbassat & Sipper, 2011) apply GP to zero-sum, deterministic, full-knowledge board games. (Alhejali & Lucas, 2010) used GP to evolve a variety of reactive agents for a simulated version of Ms. Pac-Man. The work presented by (Hong & Zhen Liu, 2010) presents a GA based-Evolvable Motivation Model for bots. Both the works of (Mora et al., 2010a) and (Mora et al., 2010b) focus on the adoption of GAs and GP techniques to evolve bots' behavior in Unreal. Last but not least, (Wong & Fang, 2012) study the applications of neural network and GAs techniques for building controllers for automatic players.

## 2.2 GAs and commercial video games

The use of GAs in commercial video game is still quite limited, at least as far as developers disclose it. The three major titles whose gameplay relates heavily on evolution are *Creatures* – released in its first version by Millennium Interactive in 1996 (Fig. 1), *Spore* – released by Maxis in 2008 (Fig. 2), and *GAR: Galactic Arms Race* – released by Evolutionary Games in 2010 (Fig. 3). None among them uses GAs in a way similar to that envisaged by the present work. In particular, *Creatures* only seems to use chromosome to generate minor variations (e.g. skin colour) in the different generations of Norms (the fictional creatures populating the game world), while the evolution of the specie is based mainly on learning and reasoning algorithms. *Spore* exploits GAs to produce automatically animations for the species created by the players, and uses extensively procedural generation for “evolving” content pre-made by developers. Nonetheless, the game main focus is on the evolution of a single user-created specie. *Galactic Arms Race* exploits NEAT – NeuroEvolution of Augmenting Topologies algorithms to develop spaceships' weapons accordingly to the player's play style.

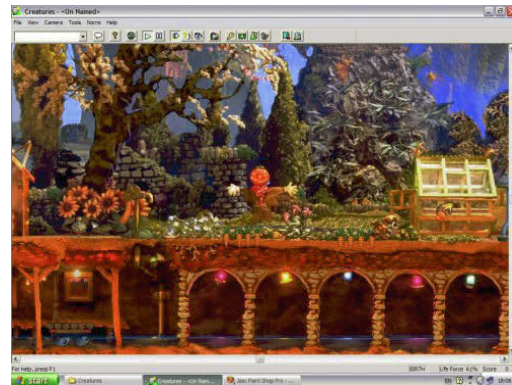


Figure 1- Screenshot from the video game *Creatures*



Figure 2 - Screenshot from the video game *Spore*



Figure 3 - Screenshot from the video game *GAR*

## 3. Genetic Algorithms

Genetic Algorithms (GAs) are a particular class of algorithms applied to solve many classes of problems, mainly belonging to the Artificial Intelligence (AI) field, but – more in general – they are useful in many optimization problems and in heuristic search processes. They have been inspired by Darwin's evolution theory: the chromosomes of a set of individuals represent a population, and a new generation in the population is produced by recombining, according to specific rules, the genetic material. For each generation, an ad hoc *fitness*



function selects the most “suitable” parents and iterate the reproduction process on them. To produce “children” the algorithm operates genetic recombination techniques (*crossover*) and *mutations* – similarly to what happens in the natural world – on chromosomes represented by bit sequences. Once the new generation has been created, the algorithm verifies whether the population registers any improvement in any relevant feature. If yes, parents will be discarded and their sons will substitute them in the reproduction process. Generally, these steps will be iterated until some optimal solution is reached (Mitchell, 1998; Mode & Sleeman, 2012; Koza, 1998).

### 3.1 Crossover

GAs use *crossover* to mix the genes of the two parents. Crossover can be achieved by different approaches:

1. *Single point crossover*: the chromosome is split into two parts in a randomly selected point. The chromosome describing the offspring is composed by the first section of the first parent chromosome and by the second part of the second parent chromosome. A second offspring will inherit the remaining two parts;
2. *Two points crossover*: the chromosome is split into three parts, the first and third part of the first parent plus the second part of the second parent become the genetic heritage of the first whelp, while the remaining parts go to the second whelp;
3. *Uniform crossover*: this approach provides a higher genetic variation, since each gene of the whelp is copied – randomly – from one of the corresponding gene belonging to one of the parents. The second whelp will have the genes not chosen for creating its “brother” chromosome;
4. *Arithmetic crossover*: the offspring chromosomes are the results of some arithmetic operations on the parents’ genes (e.g. an AND operation).

For our aim is to generate the broadest diversity among monsters, we have opted for the *uniform crossover* approach. Actually, the first two techniques (*single* and *two point(s) crossover*) provide a limited variation in the genetic heritage of the offspring, while the *arithmetic crossover* is not suitable for the structure we have used to represent chromosomes (see §4).

### 3.2 Mutation

Genetic *mutation* is useful for inserting into the offspring’s chromosome some characteristics not inheritable from parents, since not present in their genetic heritage. Similarly to what happens in nature, mutations can introduce a new characteristic or modify/destroy an existing one.

In our work we have consider only the possibility to create/destroy characteristics, since modifying an existing one (e.g. the beak of a bird that increases its length from generation to generation, according to Darwin’s theory), besides requiring more complex data structures to be

managed, would not have made much sense in a typical MMOs. Moreover, representing such a mutation would create more than one headache to someone in charge of developing graphics 3D models and animations for the monsters.

## 4. Selecting and describing monsters

To provide a significant population of monsters, we have tried to describe them accordingly to the characteristic and skill structures used in the most popular MMOs and MORPGs (such as World of Warcraft). Since these games can be set in very different periods (ranging from the most remote past, to the far future), also the types of mobs present can vary widely, and may require a different representations of their genes. This implied to make some choices, such as selecting an historical setting and sticking to it.

Since the vast majority of successful video games are based on a *fantasy* setting, we have opted for it. This, inevitably, led to lose some generality in the monsters representation, but on the other side offered the advantage to guarantee coherence and meaningfulness. Moreover, a generative algorithm based on fantasy setting, probably is more likely to be inserted into a video game. Finally yet importantly, such an exploited setting offers a greater variety of ready-to-use monsters.



Figure 4 - Fantasy mobs: an Ent and a Beholder

An accurate description of mobs (e.g. numerical values for their skills) is generally difficult – or quite impossible – to extrapolate from commercial MMOs. Luckily, these games are generally more or less sophisticated derivations of paper-and-pen Role Playing Games (RPGs), such as the renowned Dungeon and Dragons (D&D) (Bartle, 2003). This offered us the opportunity to exploit the huge corpus of information about physical aspect, characteristics and skills present in the manuals of tabletop RPGs games. In particular, the description of the monsters population we have adopted is based on the mobs described in the Dungeons and Dragons (D&D) manuals: (Dungeons and Dragons, 2000, 2003). We have analyzed more than 150 fantasy monsters (see Fig. 4 for an example), and we have created a “candidate” list of monsters that could have been represented and used as input for a GAs. We have not included in the list both immortal (e.g. angels, quasi-divine creatures, etc.) and undead (e.g. zombies, vampires, ghosts, etc.) monsters: the first ones could have caused overpopulation, while the second ones usually do not reproduce (at least in a “natural” way).

Characteristic	Type	Range
Sex	physical	2
Head number	physical	0-7
Arms number	physical	0-8
Legs number	physical	0-8
Eyes number	physical	0-8
Skin colour	physical	0-8
Eyes colour	physical	8
Tail number	physical	0-8
Wings	physical	yes/no
Fins	physical	yes/no
Gills	physical	yes/no
Scales (fish-like)	physical	yes/no
Scales (dragon-like)	physical	yes/no
Feathers	physical	yes/no
Hair/fur	physical	yes/no
Size	physical	8
Type (animal, magical creature, etc.)	nat. ab.	8
Movement: swimming	nat. ab.	yes/no
Movement: flying	nat. ab.	yes/no
Movement: digging	nat. ab.	yes/no
Movement: climbing	nat. ab.	yes/no
Breathing: air	nat. ab.	yes/no
Breathing: water	nat. ab.	yes/no
Breathing: fire	nat. ab.	yes/no
Breath and type (e.g. fire, ice, etc.)	mag. ab.	6
Natural weapons and type (e.g. claws)	mag. ab.	8
Aura and type (e.g. fire, etc.)	mag. ab.	7
Casts spells	mag. ab.	yes/no
Immunity ad type (e.g. to poison)	mag. ab.	6
Lycanthropy	mag. ab.	yes/no
Shapeshifter	mag. ab.	yes/no
Fast healing	mag. ab.	yes/no
Regeneration	mag. ab.	yes/no
Resistance to spells	mag. ab.	yes/no
Resistance to dispel	mag. ab.	yes/no
Damages reduction (e.g. thick skin)	mag. ab.	yes/no
Poisonous	mag. ab.	yes/no

Table 1 - Genes of the main mob's characteristics

Each monster is described by a chromosome, which maps its characteristics and skills. Characteristics can be: *physical* (e.g., number of legs, eyes colour, etc.), *natural abilities* (e.g., breathing water, ability to swim, etc.), and *magical abilities* (e.g., immunity to fire, ability to cast spells, etc.), as detailed in Tab.1. The maximum number of variations for each characteristic has been fixed to 8 (i.e., a monster cannot have 9 or 10 arms), thus constraining the possible mutations, in order to guarantee the algorithm performances.

Monsters' characteristics can be clustered into three main groups, each of which has a different relevance or goal in the reproduction process:

- characteristics that must be different in candidate parents (e.g., *sex*);
- characteristics that must be present in order to create a living monster (e.g., *type of breathing*, *number of legs* and *heads*, etc.);
- optional characteristics, useful to better define the monster (e.g., *resistance to spells*, *lycanthropy*, etc.).

The whole complex of these characteristics can be described using 37 genes, which are the basic building blocks of each monster chromosome. The dimension of each gene depends on how many *disjunctive* variations it can express. When the gene represent a non-disjunctive characteristic (e.g. the mobs breaths both air and water, like in the case of the mermaid), it has been duplicated: this solution allows the offspring to inherit none, one or both the characteristics.

Moreover, 16 more characteristics (*generation*, *current generation*, *challenge rating*, *life*, *strength*, *dexterity*, *constitution*, *intelligence*, *wisdom*, *charisma*, *armour class*, *speed*, *attack*, *reflexes*, *temper*, *will*) are necessary. Each of them is described by a number, since they may assume values too big for being easily managed by binary code: for example, the “*generation*” characteristic (which represent the maximum number of generations to which the monster is allowed to survive – i.e. it is a proxy for its lifespan) may ideally vary between 1 and infinite. As a consequence, any monster used in the GOLEM project is represented by a chromosome composed by 53 genes (37 basic plus 16 special). To notice that not all the characteristics used to describe monsters in D&D have been mapped into the chromosome, since several among them are not relevant for a MMO and/or cannot be properly managed through reproduction (e.g. *horse-riding* ability).

## 5. Creating and balancing a monsters population

Usually, GAs are used to solve optimization problems. In our work, instead, we have investigated to which extent GAs can be fruitfully exploited to generate a variety of new and unpredictable monsters for MMOs, in order to increase fun for players. To obtain this effect, our algorithm, that we have called GOLEM (Generator Of Life Embedded into MMOs), selects a male and a female candidate parent, mixes up (using *uniform crossover*) their genes, produces some mutations and creates two whelps, each of which possesses a subset of the parents' genome, plus a possible mutation. At this point, the algorithm ends, without applying any optimality criterion.

To implement GOLEM, besides creating a proper structure for representing the chromosome, we had to deal with some other design problems. Namely:

- choosing the number of whelps produced by each couple in each generation;
- defining some parameters to verify whether freshly created species are suitable for survival;
- balancing the population numerosity.

### 5.1 How many whelps?

Choosing how many whelps each couple of parents should generate in each generation has been a crucial point in the design of GOLEM. We ended up in deciding to generate *two* puppies. Actually, producing only one whelp implied a certain probability that several characteristics would have disappeared (since casually never transmitted to the offspring), thus de facto partially nullifying the effect we wanted to obtain (maximizing *variety*). On the other side, one single puppy would have meant a linear increase in the population numerosity and a better scalability of the software application. At the opposite end, a large number of puppies guarantees that practically every possible characteristic is preserved in the offspring, but also causes a steady and rapid increase in the population, which would reach its upper limit in a bunch of generations. As a result, we would end up with a group of individuals close relatives and presenting quite similar characteristics. For example: a starting population of 30 mobs, producing 8 whelps for couple in two interactions per generation, with an upper bound of 78 individuals for the population, would saturate in 3 generation. As a consequence, only 12 individuals have had the possibility to interact (around the 40% of the starting population).

### 5.2 Are new species suitable for survival?

Crossover and mutation could produce a mob with characteristics that could diverge significantly from those of its parents. Since the mob should be inserted in a game world, it is necessary to verify its “credibility”. In particular, we want to be sure that the newly created specie is able to survive in its *habitat*. Think, for example, to the offspring of a mermaid and a human: it could be born on the mainland, but be suitable only for marine life. In the GOLEM project we have considered four possible habitats: *forests*, *mountains*, *lava flooded areas* and *water*. Whelp’s characteristics that must be checked against its habitat to verify its survival probabilities are: *breathing* (e.g. a mob breathing water will not survive in a forest) and *movement type* (e.g. a mob only able to swim, will not survive on a mountain). To notice that several monsters could have more than one respiratory system (e.g. mermaids), thus they suffer some *malus* when outside their primary habitat, but they do not die. Moreover, some other characteristics may provide a *bonus/malus* to the whelp according to the surrounding environment: fins, gills, scales (both fish and dragon-like), feathers and fur.

### 5.3 How many individuals in the population?

Although some whelp will die due to unsuitability to the environment, it is necessary to keep a control on the population numerosity, in order to avoid overpopulation. This implies defining an upper bound to the number of individuals

alive, and some criteria to “kill” several mobs when their number reaches a certain threshold. For this reason, we have introduced both death *by old age* and *by chance*.

**5.3.1 Death by old age.** In this case, for each generation and mobs, GOLEM verifies whether the mob has reached the end of its lifespan (in terms of number of generations to which it can survive). In particular, each monster has a *longevity* that can assume one value among *short*, *medium* and *long*, to simulate the different lifespans of different monsters.

**5.3.2 Death by chance.** In a game world, a mob can die also because it has been killed by a player (and generally this is one of the main goal of players!) or, in few cases and under special game design conditions, due to illness. To avoid under populating the world, GOLEM applies casual death only when the population numerosity reaches the 70% of its maximum. This threshold has been defined by trials. To decide whether a mob should die by chance, we have grouped monsters into four categories, according to their “*challenge rating*” (a numerical value representing a proxy of the monster hardness and stamina, usually adopted in large number of RPGs): stronger mobs get killed with a lower probability. Once the challenge rating level has been (randomly) chosen, an individual in that category is randomly extracted to die.

## 6. The GOLEM algorithm

The GOLEM algorithm takes as input a starting population (whose numerosity grows constantly, generation after generation), and executes four main functions: *selection*, *crossover*, *mutation* and *evaluation*, after which a new generation is added to the population. All the functions have been implemented using C++, since it is one among the most diffused languages in the video game industry.

### 6.1 Selection

This function main goal is to select couples of candidate parents. Its first action is verifying whether some mobs should die by old age or by accident. That is to say, firstly GOLEM compares each mob lifespan against the current generation number, and then verifies if the population numerosity has reached the 70% of its maximum value. If yes, death by chance occurs as described in §5.3.2. Once this verification is concluded, the function selects candidate parents: if the “challenge rating” of the two mobs in a couple is too different, they do not mate. This check simulates the fact that, in a fantasy world, powerful mobs (e.g. a dragon) generally are uninterested in mating with weaker creatures (e.g. a kobold). The challenge ratings of our mobs have been grouped in four main classes: *low*, *medium*, *high*, *invincible*.

### 6.2 Crossover

The function takes as input the chromosomes of the parents selected by the Selection function and operates a *uniform crossover* (§3.1) on them. The chromosomes of the two resulting whelps are randomly filled with their parents’ genes.

### 6.3 Mutation

Once we have the offspring chromosomes, we apply *mutation* in order to increase variety and avoid the possibility of obtaining a population too uniform. GOLEM mutation changes randomly the value of *one* (and only one) randomly chosen characteristic; the mutation happens in the 90% of the cases, in order to add variation in the population. The possibility to generate randomly exactly the starting value for a characteristic is not excluded; hence, obviously, the probability to have an actual mutation increases with the length of the string representing the specific characteristic. Lethal or impairing mutations have not been introduced, since it would not have made any sense in a population of mobs for a video game.

### 6.4 Evaluation

It is now necessary to evaluate whether whelps can survive the habitat were they are born, as described in §5.2. If yes, the function checks if some *bonus/malus* applies: e.g., a water-born mob devoid of fins and scales can survive in the sea, but is weakened; as a consequence its skills (which are represented by numerical values – see §4) will suffer a 20% *malus*. The evaluation is applied to every whelp created in the current generation.

### 6.5 Ending condition

Generally GAs end when a certain condition – which determines the “optimum solution” – is met. The GOLEM algorithm is not intended for optimization, nor it has to evaluate the quality of the offspring (our goal is to maximize diversity), hence its ending condition is met when it reaches a predefined number of generations.

### 6.6 GOLEM fine tuning

In the definition of GOLEM we had to face several criticalities. In particular, we had to fine-tune by hand the value of several parameters in order to obtain useful results (in terms of mobs generated) and good performances. As a matter of fact, the outcomes generated by the algorithm are affected by:

- the number of characteristics represented by the chromosome;
- the probability to have a mutation;
- the probability to have death by chance;
- the probabilities, for a low/medium/high/invincible *challenge rating* mob, to be chosen for death by chance (respectively 0.8, 0.6, 0.3 and 0.1);
- the maximum numerosity of the population;
- the number of generation to produce;
- the number of whelps for each mating.

## 7. GOLEM primary results

As an example of what can be obtained from GOLEM, let's see what happens when crossing two very different monsters: a griffin and a goblin (see Fig.5). By recombining and mutating their chromosomes, we obtain two whelps, whose

genome differs from those of their parents respectively 18.8% and 28.3% (see Tab.2 – not binary values in greyed cells).



Figure 5 - A griffin and a Goblin

To estimate the level of diversity among generations, the variation should not be calculated taking as reference the starting couple, since offspring genes, after a certain amount of generations, could – by chance – configure (partially or completely) exactly as those of the ancestors. Consequently, these variations would not be included into the total amount of variations. Hence, the degree of diversity provided by the algorithm should be evaluated taking into consideration the differences between each couple of two subsequent generations. If we adopt this approach, we can notice that, with a starting population composed by 2 individuals, after 10 generations, the average genetic difference among two subsequent generations is of 7.1 genes (that is to say the 13%), while after 20 generations this value increases till the 7.15 (13.4%). This value then decreases in the following generations till it becomes constant, due to an increasing similarity in the genetic heritage (see Fig. 6; note that, since the comparison is made on only 1 puppy, the figure shows only half the generations on x axis). After 100 generations, the difference between parent and son is only of 1.9 genes (3.5%).

If the starting population increases to 18 different individuals, the algorithm performs better. As Fig.7 shows, the difference among contiguous generations is of 7.4 genes (13.9%) on average, and, even after 100 generations, the genetic mixing is still substantial.

After a dozen of tests, we have established – by trials and errors – that GOLEM performances are at their best in the following situation:

- population starting numerosity: 18;
- maximum population: 200 individuals;
- number of generation: 100;
- number of whelps per generation: 4 or 8;

in this case, the final numerosity of the population is 170, with a good diversity among mobs.

In the case a larger number of monsters is required, a quite good performance is provided by this configuration:

- population starting numerosity: 18;
- maximum population: 400 individuals;
- number of generation: 150;
- number of whelps per generation: 4 or 8;

in this latter case, the final numerosity of the population is 330, with a quite good diversity among mobs: approximately 30 individuals have a very similar genome (that is to say the difference is no more than 2 genes), thus only the 10% of the population is represented by very similar – or even identical – monsters. A further increase in the population numerosity causes the appearance of too many very similar chromosomes.



Characteristic	Goblin	Griffin	Whelp 1	Whelp 2
Max. generations	20	30	30	20
Current generation	1	1	1	1
Challenge rating	1	4	1	1
Life (hits)	5	59	59	59
Strenght	11	18	11	11
Dexterity	13	15	15	15
Constitution	12	16	16	16
Intelligence	10	5	5	10
Wisdom	9	13	9	13
Charisma	6	8	6	8
Armour class	15	17	15	15
Speed	9	9	9	9
Attack	1	7	7	1
Reflexes	1	7	7	7
Temper	3	8	3	3
Will	0	5	5	0
Sex	1	0	1	0
Breathing: air	1	1	1	1
Breathing: water	0	0	0	0
Breathing: fire	0	0	0	0
Head number	0000	0000	0000	0000
Arms number	0001	0000	0001	0000
Legs number	0001	0101	0001	0101
Eyes number	0001	0001	0001	0001
Skin colour	100	000	100	000
Eyes colour	100	101	101	100
Tail	0000	0001	0000	0001
Wings	0	1	1	0
Fins	0	0	0	0
Gills	0	0	0	0
Scales (fish-like)	0	0	0	0
Scales (dragon-like)	0	0	0	0
Feather	0	1	1	0
Hair/fur	0	0	0	0
Spells	0	0	0	0
Size	100	100	100	100
Type	111	011	111	011
Movement: swimming	0	0	0	0
Movement: flying	0	1	0	1
Movement: digging	0	0	0	0
Movement: climbing	1	0	1	0
Breath and type	000	000	000	000
Natural weapons	000	001	000	001
Aura and type	000	000	000	000
Immunity and type	000	000	000	000
Lycanthropy	0	0	0	0
Shapeshifter	0	0	0	0
Fast healing	0	0	0	0
Regeneration	0	0	0	0
Resistance to spells	0	0	0	0
Resistance to dispel	0	0	0	0
Damages reduction	0	0	0	0
Poisonous	0	0	0	0

Table 2 - Genes of a Goblin, a Griffin and of their offspring

Nonetheless, it is important to underline that a similar chromosome – or even identical – not necessary implies that the mobs are “the same”. For example, a population whose maximum is fixed to 250 individuals produces 218 whelps in 100 generation (4 or 8 puppies for each generation): among them only the 20% have similar chromosomes. In fact, genes describing physical traits can be described in the same way monsters that have very different skills; e.g. both a panther and a mouse have four legs, fur, a tail, and breathes air, but their strength and hazard is deeply different.

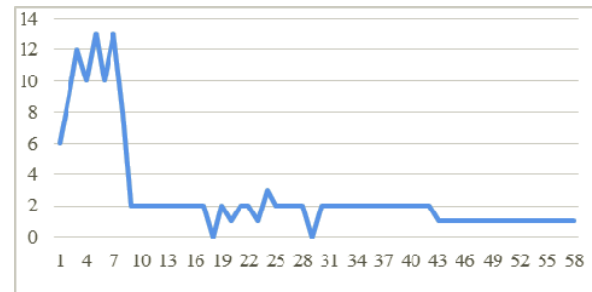


Figure 6 – Genetic difference (starting population: 2)

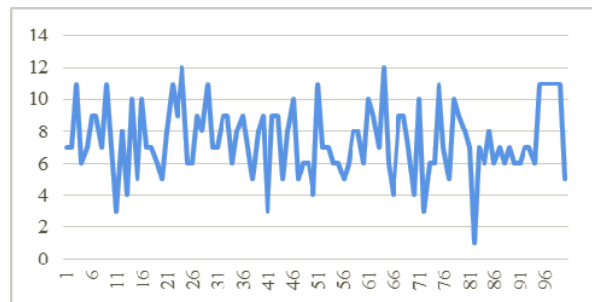


Figure 7 - Genetic difference (starting population: 18)

## 8. Conclusions and future development

In this work, we have tackled the issue of increasing diversity and unpredictability of mobs by developing an ad-hoc GA, we have called GOLEM – Generator Of Life Embedded into MMOs. In particular, we have provided a detailed general-purpose description of fantasy creatures, based on genes and chromosome, developed functions aimed at generating new species by uniform crossover and mutation on the genome of monsters parents, implementing an overall logical architecture that partially mirrors a biological ecosystem. Only several specific aspects of an actual ecosystem have been modelled: those relevant for mimicking a meaningful virtual environment for a video game. We have then tested and fine-tuned the algorithm in order to obtain outcomes – in terms of monsters population – useful for a MMOs.

The GOLEM project provides only the GA algorithm and the database of the monsters characteristics used to create the chromosomes. To be embedded into an actual MMOs – or,

more in general, in a video game –, it could, and should, be expanded in many directions. In particular, we have envisaged several possible future developments for the GOLEM algorithm:

- adding *dominant* and *recessive* genes: this development needs a careful design and a fine tuning, since it may imply a decrease in the number of characteristics actually expressed (*phenotype*) by the genetic heritage (*genotype*);
- introducing more significant mutations, which, e.g., may increasingly modify in time a specific characteristic;
- adding interdependence among specific genes: e.g. a mobs able to breath water will also have gills;
- refining the death by old age/by chance function, to better simulate monster aging (e.g. older monsters will not reproduce and will have more chances to succumb by illness).

Moreover, several refinements and simulations will also be developed to test GOLEM performances under the following conditions:

- the algorithm is embedded into a client-server architecture, designed to support a MMO, hence real-time interactions;
- the game environment is populated by players, which interact with the mobs (and kill them);
- the number of players varies (hence the population of the monsters should adapt).

Last, but not least, some corollary – but nonetheless of fundamental relevance for using GOLEM in a video game – imply such complex issues to generate several autonomous research areas:

- creating tools able to provide automatically – and in quasi real-time – graphic 3D representations and animations of GOLEM-generated monsters;
- adapting GOLEM as a basis for generating game maps directly related to the monsters characteristics;
- measuring the impact of monsters diversity generated by GOLEM on players' overall satisfaction with the game.

## References

- Alhejali, A. M., and Lucas, S. M. (2010). Evolving diverse Ms. Pac-Man playing agents using genetic programming. *Computational Intelligence (UKCI)*, 2010 UK Workshop on. IEEE, 2010.
- Barros, G. A. B., et al. (2011). An Application of Genetic Algorithm to the Game of Checkers. *Games and Digital Entertainment (SBGAMES)*, 2011 Brazilian Symposium on. IEEE.
- Bartle, R. A. (2003). *Designing Virtual Worlds*. New Riders Publishing, Indianapolis (Indiana).
- Benbassat, A., and Sipper, M. (2011). Evolving board-game players with genetic programming. *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*. ACM, 2011.
- de Carvalho, L. F. B. Silva, et al. (2010). An application of genetic algorithm based on abstract data type for the problem of generation of scenarios for electronic games. *Intelligent Computing and Intelligent Systems (ICIS)*, 2010 IEEE International Conference on. Vol. 2. IEEE, 2010.
- Dungeons and Dragons (2003). *Monster Manual: core rulebook III v 3.5*. Wizard of the Coast
- Dungeons and Dragons (2000). *Player's handbook: core rulebook I*. Wizard of the Coast
- Esparcia-Alcázar, A. I., and Moravec, J. (2012). Fitness approximation for bot evolution in genetic programming. *Soft Computing*: 1-9.
- Frade, M., de Vega, F. F., and Cotta, C. (2012). Automatic evolution of programs for procedural generation of terrains for video games. *Soft Computing* 16.11: 1893-1914.
- Frade, M., Fernandez de Vega, F., and Cotta, C. (2010). Evolution of artificial terrains for video games based on accessibility. *Applications of Evolutionary Computation*. Springer Berlin Heidelberg. 90-99.
- Halim, Z., and Raif Baig, A. (2011). Evolutionary Algorithms towards Generating Entertaining Games. *Next Generation Data Technologies for Collective Computational Intelligence*. Springer Berlin Heidelberg. 383-413.
- Hong, Y., and Zhen Liu (2010). A First Study on Genetic Algorithms Based-Evolvable Motivation Model for Virtual Agents. *Multimedia Technology (ICMT)*, 2010 International Conference on. IEEE, 2010.
- Inführ, J., and Raidl, G. R. (2012). Automatic generation of 2-antwars players with genetic programming. *Computer Aided Systems Theory-EUROCAST 2011*. Springer Berlin Heidelberg. 248-255.
- Koster, R. (2004). *A Theory of fun for game design*. Paraglyph Press.
- Koza, J.R. (1998). *Genetic programming: on the programming of computers by means of natural selection*, MIT Press.
- Maggiolini D., Nigro A., Ripamonti L.A., Trubian M. (2012a). Loot Distribution in Massive Online Games: foreseeing impacts on the player base. *8th International Workshop on Networking Issues in Multimedia Entertainment (NIME'12)* Int. Conf. on Computer Communication Networks ICCCN 2012 Munich, Germany.
- Maggiolini D., Nigro A., Ripamonti L.A., Trubian M. (2012b). The Perfect Looting System: looking for a Phoenix? *IEEE Conference on Computational Intelligence and Games (CIG)*, Granada (Spain).
- Mitchell, M. (1998). *An Introduction to Genetic Algorithms*, MIT Press.
- Mode, C.J. and Sleeman, C.K. (2012) *Stochastic Processes in Genetic and Evolution: Computer Experiments in the Quantification of Mutation and Selection*, World Scientific Publishing Company.
- Mora, A. M., et al. (2012). Effect of Noisy Fitness in Real-Time Strategy Games Player Behaviour Optimisation Using Evolutionary Algorithms. *Journal of Computer Science and Technology* 27.5: 1007-1023.
- Mora, A. M., et al. (2010a). Evolving bot AI in Unreal. *Applications of Evolutionary Computation*. Springer Berlin Heidelberg, 2010. 171-180.
- Mora, A. M., et al. (2010b). Evolving the cooperative behaviour in Unreal™ bots. *Computational Intelligence and Games (CIG)*, 2010 IEEE Symposium on. IEEE.
- Mourato, F., dos Santos, M. P., and Birra, F. (2011). Automatic level generation for platform videogames using genetic algorithms. *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*. ACM.
- Onieva, E., et al. (2012). An evolutionary tuned driving system for virtual car racing games: The AUTOPIA driver. *International Journal of Intelligent Systems*. 27.3: 217-241.
- Sorenson, N., and Pasquier P. (2010). Towards a generic framework for automated video game level creation. *Applications of Evolutionary Computation*. Springer Berlin Heidelberg. 131-140.
- Taylor, T.L. (2006) *Play between worlds: exploring online game culture*. MIT Press.
- Wong, S., and Fang, S. (2012). A Study on Genetic Algorithm and Neural Network for Mini-Games. *Journal of Information Science and Engineering*, 28.1: 145-159.

# Evolving Behaviour-Dependent Strategies in Agent Negotiations

Darius P. Falahat, Enrico H. Gerding and Markus Brede

Electronics and Computer Science,  
University of Southampton, SO17 1BJ  
dpf1g11@ecs.soton.ac.uk

## Abstract

We use genetic algorithms to evolve trading strategies for iterative bilateral negotiations between buyers and sellers. In contrast to previous work we evolve purely reactive strategies that base decisions on memories of behaviour in previous negotiation rounds. We find that simulations lead to three main types of behaviour: (i) cooperative outcomes in which bargaining leads to an agreement and equal sharing of profits, (ii) uncooperative outcomes in which negotiations are not successful and (iii) outcomes in which one party profits at the expense of the other. The frequencies of each type of behaviour vary when the probability for negotiations to terminate is changed, confirming our hypothesis that cooperation should decrease as this break-off probability increases. Comparisons of the results to tit-for-tat (TFT) strategies and previous research on the iterated prisoner's dilemma (IPD) are used to understand simulation results, and we observe the emergence of TFT behaviour during periods of agent cooperation.

## Introduction

Trading over the internet and other communication networks continues to grow ever more prevalent in both high-income economies and emerging markets. Research in e-commerce is relevant to high frequency trading, supply chain management and many areas that involve some sort of online transaction. Automated negotiation is central to many of these systems. The application of game theory to automated negotiation is well-established (Binmore & Vulkan, 1999) with practical uses in e-commerce realised early on (Oliver, 1996). The core mechanics of this type of negotiation are retained in a widely used, simple bilateral negotiation model with alternating-offers (Rubenstein, 1994). We use this framework in conjunction with evolutionary computation to make a fresh contribution in a field neglected by the literature in recent years: behaviour-dependent negotiation.

Previous work (Gerding et al., 2004) has focused on using GAs to investigate the emergence of time-dependent negotiation strategies. In the framework of Matos et al. (1998) and Faratin et al. (1997), an agent's strategy can be determined by closeness to a negotiation deadline (time-dependence), the scarcity of a diminishing re-

source (resource-dependence) and the actions of an agent's opponent (behaviour-dependence). However, regarding behaviour-dependence the authors only consider variations of TFT. The intention of this paper is to address this gap in the literature and develop behaviour-dependent negotiation. Building on previous work in the context of the IPD (Lindgren, 1992) we aim to test for the emergence of cooperative behaviour within a framework of reactive strategies.

## Related Research

Automated negotiation has been shown to be vital to e-trading (Sierra et al., 1997) and the use of GAs to identify the most successful negotiation strategies has long been widespread (Matos et al., 1998). GAs use the powerful processes observed in biological evolution: selection of the best-performing (fittest) individuals to replace a population, combined with a small probability of these new individuals undergoing mutations in order to generate diversity and explore the strategy space. GAs have been popular in many fields because they make no assumptions about agent rationality, or the fitness landscape in general. The propagation of agent strategies into new generations is purely based on their fitness.

The complexity and capability of these agent-based models has progressed over time such that, in addition to bargaining over the price of goods (single-issue negotiation), agents can argue over additional properties such as deadlines, cope with multi-issue negotiation (Gerding et al., 2000) and incomplete information (Fatima et al., 2004). This model attempts to take a simple approach that only involves bargaining over a price for the goods; fitness is simply defined in terms of agent utility.

In the context of iterated social dilemma games like the IPD, the situation we consider has been studied extensively by the artificial life community. The prisoner's dilemma is a classic one-shot game, where cooperation rewards higher utility but the rational choice is to defect. In the IPD agents play each other repeatedly, which introduces the potential for more complex behaviour, such as punishment for defection in previous games. Theoretically for a finite number of

games, backwards induction implies that the rational choice is to defect each time, although this reasoning does not hold if the number of negotiation rounds played is uncertain and not a priori known to agents. When strategies were tested against each other in an IPD tournament (Axelrod, 1980), the most effective and robust strategy was for an agent to cooperate on the first move and then retaliate with the opponent's previous action. This strategy, TFT, was successful because it rewarded cooperation and punished defection. It should be noted that with the IPD there is a payoff at the end of every negotiation round, whereas in the bargaining model we consider there is only a chance of a payoff at the end of a full negotiation between two agents. Hence results from the IPD can not be translated directly to the simulation results we present in this paper.

### Contributions

We have seen that most related research on bilateral negotiation has focused on time-dependent strategies. Our main contribution is to remedy the lack of behaviour-dependent research: the actions of the agents in this model depend entirely on their own past behaviour, and their opponent's past behaviour. The approach we take is similar to Lindgren (1992), where the author models a single population of agents playing the IPD against each other. Lindgren allows the mutations to make the agent strategies more complex, and finds that selection favours cooperative strategies. This exploration of the strategy space via mutations allowed the author to observe extinctions, periods of stasis and other phenomena. Our research also uses a one-population model, but the agents interact via the bilateral negotiation method instead of the IPD. We define cooperation as the sharing of profits equally, meaning exactly equal utility for both agents. Furthermore mutations are only used to randomly select among existing strategies, not to change the size of agent chromosomes. This means the strategy space remains unchanged throughout a simulation, unlike in Lindgren (1992) where the author allows the strategy space to increase.

Despite using a simple negotiation framework, the model produces agent strategies that can be compared to well-known results from the literature (see Results section). The extensibility of the model means the existing framework can be easily built up to further complement the automated negotiation literature from a behaviour-dependent perspective.

### Overview

The following section will describe the model in detail, first tackling the negotiation framework and then explaining how the GA works. The section after that reports the results, stating the experimental setup and the parameters used in the simulations. We then present and discuss the observed types of agent behaviour, including analysis of a frequency distribution showing how the different classes of evolved behaviour change over time. The final section summarises the

paper including the results, and we suggest several avenues for future research.

## Model Description

The model can be understood best by treating it as two distinct components: a negotiation framework and a GA. This section will describe how these components work in detail (see Figure 1 for an overview). The negotiation framework has the bilateral alternating-offers protocol at its core, but the discussion will also include a description of agent strategies and their time-independent nature. The GA is essentially a search algorithm that is applied to the framework; it finds the best performing agents and ensures there is a high probability of them being passed onto the next generation.

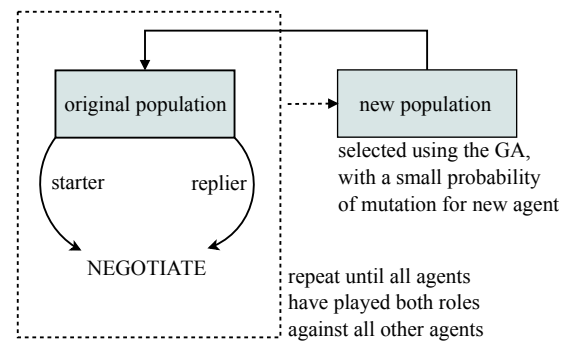


Figure 1: An overview of the interaction between the negotiation framework and GA. One full circuit of the diagram comprises a single *generation*, i.e. the complete interaction and replacement of an agent population.

### Negotiation Framework

The bilateral negotiation protocol involves two agents bargaining over goods. The mechanics of how they make offers, counter-offers and contemplate agreement varies considerably over the field of research. In this paper we use a specific protocol with a small strategy space, which keeps analysis relatively simple.

Negotiation between two agents means a Buyer agent and a Seller agent proposing offers and counter-offers to each other, in an attempt to agree on a price for the item. The Buyer initiates proceedings and the Seller replies with a counter-offer. During a negotiation, the Buyer will always begin by offering 0 for the item, while the Seller will make an initial offer of 10. An agent's reserve price is defined as his opponent's initial offer, so the price of the goods stays between the range  $[0, 10]$  even though there are no explicit constraints on its value. To simplify matters, the agreed price is converted to utility in the following way. If the Buyer accepts the Seller's offer,



$$U^b = O_{s \rightarrow b}^t \quad (1)$$

$$U^s = 10 - O_{s \rightarrow b}^t \quad (2)$$

where  $U$  is an agent's utility, and  $O_{s \rightarrow b}^t$  is an offer from the Seller ( $s$ ) to the Buyer ( $b$ ), at negotiation round  $t$ . Hence utility for both agent types follows the same convention. An inability to come to an agreement results in both agents being punished with zero utility.

There are only two actions, concede and non-concede, available to agents in this model because only two are necessary for bilateral negotiation. Including further actions would be interesting but an area for future work; a spartan philosophy was used for the base model in an attempt to reduce unnecessary complexity. The actions are the same for Buyers and Sellers, but the concede action has different outcomes depending on the agent.

For the Buyer the concede ( $C$ ) and non-concede ( $N$ ) actions are as follows,

$$C : O_{b \rightarrow s}^{t+1} = O_{b \rightarrow s}^t + 1, \quad N : O_{b \rightarrow s}^{t+1} = O_{b \rightarrow s}^t \quad (3)$$

While for the Seller,

$$C : O_{s \rightarrow b}^{t+1} = O_{s \rightarrow b}^t - 1, \quad N : O_{s \rightarrow b}^{t+1} = O_{s \rightarrow b}^t \quad (4)$$

This incremental approach to modifying offers is used instead of other approaches, such as making a complete offer every time, because it restricts the strategy space and limits the complexity of the model.

The alternating-offers protocol is sequential in nature, which means the Buyer will make an offer, followed by a counter-offer from the Seller. The Seller compares the utility it could get by accepting his opponents offer, to his own counter-offer. If the Buyer's offer is more favourable, the Seller will accept. Otherwise, the Seller will decline and the process will continue. The negotiation method  $M$  is described in more detail in Eq. (5) from the perspective of a Buyer  $b$  who has received an offer  $O_{s \rightarrow b}^t$  from a Seller  $s$  at round  $t$ ,

$$M^b(O_{s \rightarrow b}^t) = \begin{cases} Quit & \text{if } t > t_{max} \\ Accept & \text{if } U^b(O_{s \rightarrow b}^t) \geq U^b(O_{b \rightarrow s}^{t+1}) \\ O_{s \rightarrow b}^{t+1} & \text{otherwise} \end{cases} \quad (5)$$

where  $t_{max}$  is the deadline. The notation used to describe the bilateral negotiation protocol is borrowed from (Fatima & Wooldridge, 2002). To lessen the impact of a hard deadline on the negotiation mechanics, the model uses the equivalent method of a break-off probability. This means there is a very small chance of the negotiation ending in each round.

Another motivation for using break-off probabilities is that this removes defection via backward induction (see Related Research) as a rational choice, because agents do not know how long a negotiation will last in advance. Using a break-off probability  $p$ , we have the expected utility,

$$\langle U \rangle = U \left[ 1 - (1 - p)^T \right] \quad (6)$$

where  $T$  is the number of negotiation rounds until agreement. The break-off probability  $p$  can be varied: this is likely to have an effect on cooperation if cooperative strategies evolve. Raising the break-off probability limits the time left to negotiate, which would mean fewer strategies can lead to equal sharing, lowering chances for cooperation to develop. We would therefore expect to see less cooperation at higher break-off probabilities, and an increased likelihood of negotiations that end with no agreement at all.

The novel extension of this model involves each agent having a memory consisting of its own and its opponent's previous offers. These memories are central to the strategies that define agent behaviour. An agent's strategy is a mapping of every possible memory to the actions concede ( $C$ ), or non-concede ( $N$ ). These actions correspond to those defined in (3) and (4). Every agent's strategy holds the information shown in Table 1 below.

		Initial		Main			
Buyer	Memory	$S$		$C, C$	$C, N$	$N, C$	$N, N$
	Action	$C$ or $N$					
Seller	Memory	$C$	$N$	$C, C$	$C, N$	$N, C$	$N, N$
	Action	$C$ or $N$					

Table 1: A representation of how agents' strategies are encoded. The order of memories is important:  $C, N$  means an agent conceded and his opponent did not.  $N, C$  means the opposite.

It should be noted that every agent carries a Buyer and Seller genome, as in Table 1. These genomes are separate: the Buyer genome is used when the agent plays as a Buyer, and similarly for the Seller genome. In a single generation, every agent will play once in both roles. An agent's utility is calculated as the average of the utility from its roles as a Buyer and Seller.

In the first round of a negotiation, the Buyer has no real memories to base an action on. So the special initial  $S$  memory is used by the Buyer, only for the first round. In the second round, the Seller does not have a full memory, only a memory of its opponent's move. This means the Seller needs its own initial special case, which depends on what the Buyer did. After the two initial offers, an agent's decision depends on any combination of its own last move, and its opponent's.

Throughout this paper we assume that agents have a one-step memory. This means agents only remember the pre-

vious round. A strategy with two-step memories would be much longer: the first two possible memories in the main body would be  $(CC, CC)$ ,  $(CC, CN)$ , and so on.

### Genetic Algorithm

Selection of the fittest agents (those with the highest utility) is achieved using fitness proportionate selection, also called roulette wheel selection in the literature. Unlike some other types of GA, there is still a small probability of choosing less fit agents. This is important because a strategy that is weak against particular strategies may be strong against others, making it slightly more realistic than completely discarding unfit strategies as with truncation selection. See the algorithm below for pseudocode showing where the GA fits in relation to the negotiation framework and the rest of the model.

---

**Algorithm 1** A simplified representation of the one-population model and GA.  $n_{max}$  is the agent population and  $x_{max}$  is the number of generations for which the simulation continues. Offers, counter-offers and utility calculations are made in the NEGOTIATE subroutine. All agents play each other in both Buyer and Seller roles, but are not allowed to play against themselves.

---

```

Require:  $n_{max}, x_{max}$ 
while  $n < n_{max}$  do
   $agents \leftarrow \text{INITIALISEAGENTS}()$ 
  while  $x < x_{max}$  do
    if  $\text{AGENTSPLAYED}() = \text{False}$  then
       $buyer, seller \leftarrow \text{PICKAGENTS}(agents)$ 
       $\text{NEGOTIATE}(buyer, seller)$ 
       $\text{RESETMEMORIES}(buyer, seller)$ 
    end if
     $\text{GETUTILITY}(agents)$ 
     $agents \leftarrow \text{SELECTION}(agents)$ 
  end while
end while

```

---

Agents also have a small chance of undergoing single-gene mutation during the selection process, and are given a smaller probability of being completely replaced by a new random agent, to properly explore the strategy space without having to run extremely long simulations. Gene mutations are simply the possibility of an action in an agent's strategy to switch randomly. For example, a Buyer's initial move could change from  $N$  to  $C$ . If the Seller's two initial moves (see Table 1) are identical, the change will make no difference to the negotiation. On the other hand if they are different, it could change the effective behaviour completely. Since mutations can affect genes that are not used during negotiations, it is possible that mutants could invade populations via neutral drift. This is a possibility because certain types of behaviour use very few genes of an agent's genome (e.g. Table 2 in the Results section), making them

potentially vulnerable to drift.

## Results

In this section, the different types of agent behaviour are categorised and the change in strategies, and agent utilities, over a simulation are plotted. The stability of the distinct negotiation scenarios are analysed, and some specific strategies are discussed in depth. Finally, the relationship between cooperative behaviour and the break-off probability is plotted and explained.

### Experimental Setup

The key parameters for the following simulation results (unless specified otherwise) are given below :

- Population size 100.
- Simulation length of 2000 generations.
- Mutation rate of  $10^{-5}$  for every gene per generation, and  $10^{-4}$  for the mutation of an entire agent. These values were arrived at by slowly decreasing from a large mutation rate, to reach a point where noise from mutations did not dominate the system. For example, there is approximately one gene switch per population every 10 generations, and the introduction of a new randomised agent happens roughly a few times over 100 generations.
- Single time-step memories, i.e. agents only remember the previous negotiation round.
- A break-off probability of  $p = 0.005$  (see Eq. (6)) is used for the simulations shown in Figures 2, 3 and 4. This probability gives agents 200 negotiation rounds on average to come to an agreement. Figure 6 uses a larger break-off probability of  $p = 0.05$  and the parameter is varied in Figure 5.

### Strategy Analysis

Three main negotiation outcomes have been observed in the model. We define these as follows:

- **Domination** is the category of outcome where one agent type has finished a negotiation with a higher utility than the other.
- **Cooperation** is the label for negotiations that finish with both agents walking away with equal utility, if agent utilities are above zero.
- **Zero Utility** outcomes are when both agents finish the negotiation with no utility. This outcome represents a failure to negotiate successfully, so it is treated as distinct to co-operation.

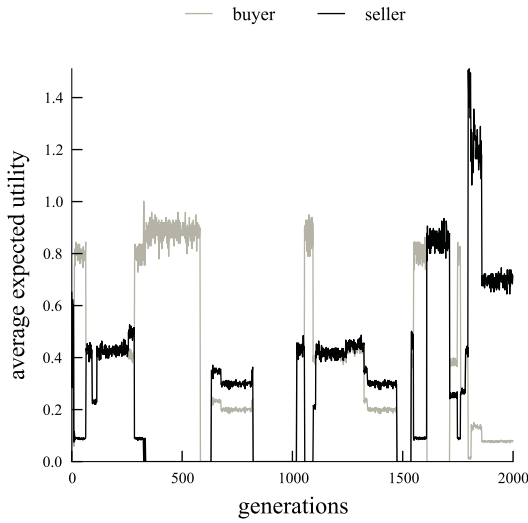


Figure 2: A long simulation run showing the change in agent utilities over time, and the volatility caused by the persistent instability of strategies.

A long simulation using the parameters set out above is shown in Figures 2 and 3. In our analysis of the negotiation outcomes we use two types of plot to clarify the agent dynamics. Figure 2 is a plot showing the average expected utility against the number of generations. The utilities presented in the plot were calculated using Eq. (6) for each agent, with a different utility value for the Buyer and Seller agent types. The expected utilities are then averaged over the population every generation to produce the following plots. Since utilities can range from 0 to 10, the expected utilities will always be less than 10, and often considerably less.

Figure 3 shows the fraction of negotiation outcomes at every generation. Every negotiation is classified as one of the three categories mentioned previously. This type of plot is more sensitive to the strategy dynamics, while the utility plots give a clearer idea of how agents are being selected over the generations.

The most obvious property of Figure 3 is that the simulation never reaches a stationary point, i.e. the strategy fractions never stabilise. In situations where the zero utility (total defection) strategy is in the majority, this lack of stability could be because these agents are scoring no utility, so the population can be invaded by mutants that use any other strategy because no strategy can perform worse. This behaviour highlights a difference to the IPD where defection clearly matters, because agents have the opportunity to punish their opponent straight away.

Invasions of these strategies consist of a build-up of mutants, followed by a swift replacement of the population as soon as utility-scoring mutants start to be selected. Although

		Initial		Main			
Buyer	Memory	<i>S</i>		<i>C, C</i>	<i>C, N</i>	<i>N, C</i>	<i>N, N</i>
	Action	<i>N</i>		<i>C</i>	<i>N</i>	<i>C</i>	<i>N</i>
Seller	Memory	<i>C</i>	<i>N</i>	<i>C, C</i>	<i>C, N</i>	<i>N, C</i>	<i>N, N</i>
	Action	<i>N</i>	<i>N</i>	<i>C</i>	<i>N</i>	<i>C</i>	<i>N</i>

Table 2: A strategy table corresponding to the most stable kind of zero utility strategy. Due to becoming locked in a cycle, only the actions in bold are taken by the agents.

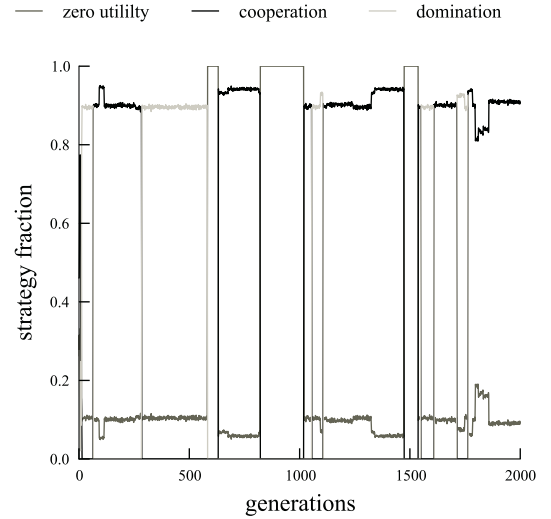


Figure 3: A plot showing the strategies corresponding to the simulation in Figure 2, displaying the changes in agent behaviour.

zero utility strategies are all inherently unstable, some are more resistant to invasion than others: an interesting case is that of a population using the genomes in Table 2.

The most common type of behaviour after the initial stage of the simulation is cooperation. There are many types of strategy that can lead to cooperative behaviour, including all-concede strategies and TFT-like behaviour. Table 3 is an example of the emergent TFT strategies we observed during periods of cooperation.

		Initial		Main			
Buyer	Memory	<i>S</i>		<i>C, C</i>	<i>C, N</i>	<i>N, C</i>	<i>N, N</i>
	Action	<i>C</i>		<i>C</i>	<i>N</i>	<i>C</i>	<i>N</i>
Seller	Memory	<i>C</i>	<i>N</i>	<i>C, C</i>	<i>C, N</i>	<i>N, C</i>	<i>N, N</i>
	Action	<i>C</i>	<i>C</i>	<i>C</i>	<i>N</i>	<i>C</i>	<i>N</i>

Table 3: A strategy table corresponding to cooperative behaviour. Both agents employ TFT-like strategies to arrive at a cooperative outcome.

The Buyer uses a pure TFT strategy, as it initially cooperates and thereafter responds with its opponent's previ-

ous move (Axelrod, 1980). The Seller effectively follows the same strategy, resulting in mutual conceding until an equal agreement is reached. For the Seller's strategy to be pure TFT, its initial actions would need to be  $C \rightarrow C$  and  $N \rightarrow N$ . The TFT strategies are not stable, possibly because groups of TFT agents are vulnerable to conceders creeping into the population via neutral drift. The concept is similar to the findings described in Nowak & Sigmund (2005): once non-discriminating conceders invade a population, they are in turn vulnerable to non-conceders (defectors). This has the effect of temporarily replacing a cooperating population with a dominating Buyer or Seller population. A sharp transition of cooperation to domination (and the reverse) can be seen at certain points in Figure 3; the abrupt nature of these transitions may be due to the relatively small size of the agent population.

We have seen from the strategy tables that, during a negotiation, an agent's actions are only determined by a fraction of its entire strategy. By tracking all possible actions available to agents and comparing this to what agents are doing, we can investigate the vulnerability of a majority-conceder population and check for neutral drift. The top plot of Figure 4 shows the fraction of all possible actions in every agent's strategy. The bottom plot shows actual agent behaviour. For the first 300 generations, as cooperative behaviour increases the fraction of concede actions also increases. Shortly after 300 generations have passed, these majority conceders are exploited: this can be seen in the dip of cooperative behaviour in the bottom graph and the corresponding reduction of conceders in the top plot. An indicative example of the neutral drift effect can be seen in the period between 400 and 600 generations, where non-conceding actions increase until they represent over 70% of all actions despite the prevalence of cooperative behaviour.

The analysis so far has focused on simulations that use a low break-off probability. Figure 5 tests our earlier prediction that higher break-off probabilities, corresponding to shorter negotiations, would mean cooperation has a lower chance to evolve. The figure was generated by recording the outcome of each negotiation (cooperation because an agreement was reached, or zero utility because negotiation terminated before agreement) every 100 generations. The fraction of each outcome was calculated at the end of the simulation, and this result was then averaged over 10 simulations to account for variance.

As expected, we find that the fraction of cooperative games with a *small* break-off probability is notably larger than when a larger break-off is used. Furthermore, Figure 5 shows zero utility outcomes are not typical until the break-off probability is increased: the trend for this type of behaviour is roughly inversely proportional to the fraction of cooperative outcomes.

Figure 6 illustrates this point further, showing how the zero utility outcomes take precedence when there is less time

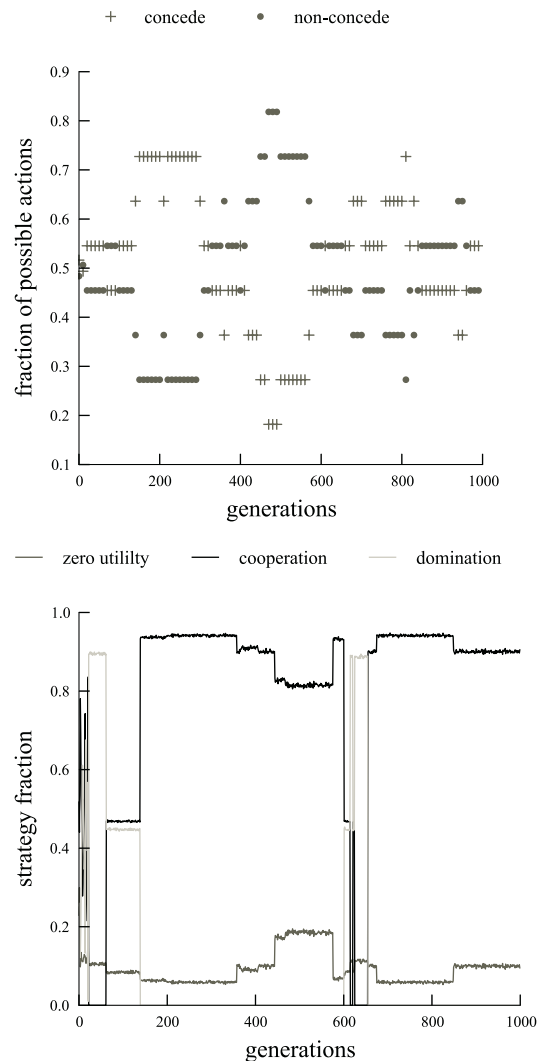


Figure 4: Top: A plot showing the fraction of possible actions for all agents over 1000 generations. Bottom: The behaviour of agents throughout this simulation.

to reach an agreement. The stark difference between Figures 3 and 6 makes sense because cooperative negotiations, which can show transient TFT-like behaviour, are often more complex (since they are discriminatory) and thus need more time to come to an agreement. Non-discriminatory strategies, such as pure defection that results in zero utility outcomes, thrive when there is less time to negotiate.

## Concluding Remarks

We built a model using evolutionary algorithms and the bilateral negotiation framework with alternating-offers protocol. Motivated by the literature discussed in the introduction and the relative lack of research on behaviour-dependent negotiation, agents were given behaviour-dependent strategies



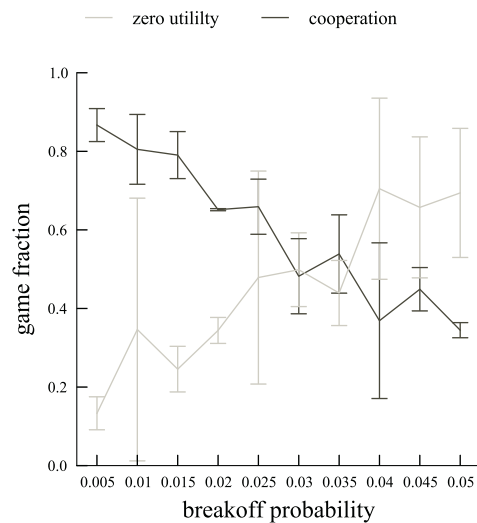


Figure 5: A plot showing the fraction of cooperative and zero utility outcomes as a function of the break-off probability. The error bars give the standard deviation for every averaged point.

that base their actions on the behaviour of their opponents. These reactive strategies were evolved and agents selected based on their performance against all other agents in the population.

Three distinct types of negotiation outcomes were observed: cooperation, zero utility and domination outcomes, although the first two types were far more common overall. Simulations proved to be unstable even when left to run for several thousand generations; this is likely due to the neutral drift discussed earlier. Analysis of agent strategies revealed the emergence of TFT-like behaviour during periods of cooperation, although direct comparisons with TFT strategies in the IPD are not possible due to the different way pay-offs are handled. In particular, the concept of *cooperation* as discussed in this paper refers to the bilateral agreement on a price for goods; this is not entirely analogous with its meaning in the evolutionary game theory literature. Finally, the assumed relationship of cooperative behaviour with the break-off probability was verified: cooperation is observed more often when negotiations can continue for a longer period of time.

Although this paper was partly motivated by research that used game theoretic concepts like the IPD, our model is not limited to applications within evolutionary game theory. The negotiation framework can essentially be used with any search algorithm in order to select the agents that perform best, not only a GA. The results reported in this paper validate our model as an effective way to investigate the emergence of cooperation in the context of a sequential, bilateral bargaining framework.

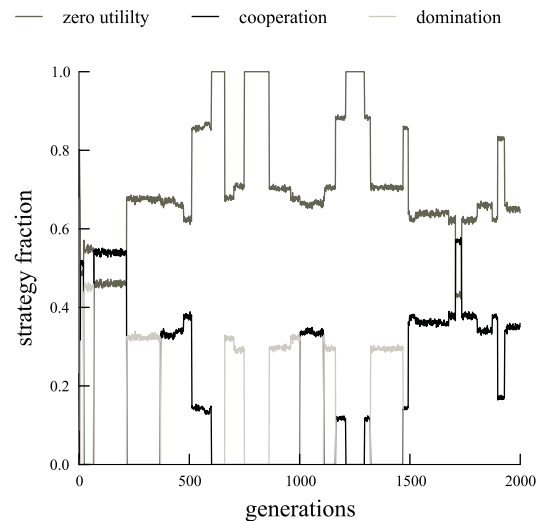


Figure 6: The change of strategy types over time, using a high break-off probability ( $p = 0.05$ ) and otherwise identical parameters to the simulation shown in Figure 3, where the break-off probability was lower ( $p = 0.005$ ).

## Future Work

In this paper we developed a general modelling framework that allows for straightforward extensions in several directions. The first priority in future work would be to verify if it is possible to produce evolutionary stable strategies in our model. One aspect worth more detailed investigation is that of longer agent memories. So far preliminary results from doubling agent memories have mainly shown an extended initial period of noise, before the simulation continues into the familiar patterns of constant strategy invasions discussed above. However, an evolutionary stable solution should not be ruled out and there may be potential in an analytic approach due to the relatively small strategy space.

There are also many ways of increasing the strategy space, such as adding a third action. An interesting possibility is to make it a random choice between the existing two actions. This would allow us to explore if deterministic strategies are more favourable when pitted against unpredictable agents. Explicitly expanding the strategy space can be done by making time a parameter of the agent strategies, making the model both behaviour- and time-dependent. Currently a strategy is a single line of mappings; introducing time would effectively add another dimension, giving a line of possible actions for every round until the deadline.

## References

- Rubenstein, A. (1994). *A Course in Game Theory*. MIT Press.
- Fatima, S. and Wooldridge, M. (2002). Optimal Negotiation Strategies for Agents with Incomplete Information. *Intelligent*

*Agents VIII*, Springer Verlag.

Binmore, K. and Vulkan, N. (1999). Applying game theory to automated negotiation. *Netnomics*, 1(1):1–9.

Oliver, J. R. (1996). A machine learning approach to automated negotiation and prospects for electronic commerce. *Journal of Management Information Systems*, 13(3):83–112.

Sierra, C., Jennings, N. R., Noriega, P. and Parsons, S. (1997). A Framework for Argumentation-Based Negotiation. *Proc. 4th Int. Workshop on Agent Theories, Architectures and Languages (ATAL-97)*, pages 177–192.

Matos, N., Sierra, C., and Jennings, N. R. (1998). Determining successful negotiation strategies: An evolutionary approach. *Proc. 3rd Int. Conference on Multi-Agent Systems (ICMAS-98)*, pages 182–189.

Faratin, P., Sierra, C., and Jennings, N. R. (1998). Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3): 159–182.

Gerding, E. H., Somefun, D. J. A. and La Poutré, J. A. (2004). Automated bilateral bargaining about multiple attributes in a one-to-many setting. *Proc 6th Int. Conf. on Electronic Commerce*, pages 105–112.

Fatima, S., Wooldridge, M. and Jennings, N. R. (2004). Optimal negotiation of multiple issues in incomplete information settings. *3rd Int. Conference on Autonomous Agents and Multi-Agent Systems*, pages 1080–1087.

Gerding, E., Bragt, D. van, and Poutre, J. La. (2000). Multi-issue negotiation processes by evolutionary simulation. *Workshop on Complex Behavior in Economics*, pages 1–28.

Axelrod, R. (1980). Effective Choice in the Prisoner’s Dilemma. *Journal of Conflict Resolution*, 24(1):3–25.

Lindgren, K. (1992). Evolutionary phenomena in simple dynamics. *Artificial life II*, 10:295–312.

Nowak, M. A., and Sigmund, K. (2005). Evolution of indirect reciprocity. *Nature*, 437:1291–1298.

# Behavior as broken symmetry in embodied self-organizing robots

Ralf Der and Georg Martius

Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany  
 {ralfder—martius}@mis.mpg.de

## Abstract

Self-organization—ubiquitous in nature—is a major challenge for both artificial life and modern robotics offering intriguing perspectives for practical applications utilized so far only incipiently. There is some progress, though, in formulating general objective functions for driving systems into self-organization (SO). Based on general principles like information maximization, these approaches are domain invariant and free of arbitrariness. However, and this seems to be a major source of concerns, if nothing is specified from outside, will SO simply make the robot an arbitrary subject that is completely unpredictable in its behaviors and thus rather a thread than a hope. The aim of this paper is to show that this attitude is not justified. Instead, we develop an understanding of what happens if the system is self-organizing, what the role of the embodiment is and how we can find clues for predicting and shaping the behavior patterns emerging in a genuine SO scenario. The approach is based on a new unsupervised learning rule staging two antagonistic activities—driving systems towards instability while preserving the physical symmetries of the system as much as possible. This leads to spontaneous symmetry breaking, the leading phenomenon of SO known from nature that has been overlooked by the robotics community so far. It is shown by a number of examples that the unsupervised learning rule induces an amazing variety of behaviors—patterns in space and time that can be interpreted as broken symmetries.

## Introduction

Self-organization (SO) is a ubiquitous phenomenon in nature and a promising challenge to the creation of artificial autonomous systems. In particular, in embodied artificial intelligence, SO may provide an essential progress in the realization of embodied control. Viewing a robot in its environment as a complex dynamical system, SO can help to let highly coordinated and low dimensional modes emerge in the coupled system of brain, body and environment. In this way, instead of being programmed for solving a specific task, the robot may find out by itself about its bodily affordances and then, in a second step, one may focus on the exploitation of the emerging motion patterns—by guiding the SO process into the directions of potential benefits.

While there are many approaches toward structural SO, in particular self-assembly, the SO of behavior still is con-

sidered more as wishful thinking than as a true and systematic approach toward autonomy. A principled way toward the SO of behavior faces essentially two challenges. One is how to organize a robotic system in such a way that it starts to self-organize its behavior. Actually, the situation in that point is not too bad. There are several approaches based on formulating objective functions (OF) for SO. In recent years, several such OFs have been proposed, ranging from the maximization of predictive information Ay et al. (2008); Der et al. (2008); Ay et al. (2012); Martius et al. (2013) or empowerment Klyubin et al. (2005, 2007); Anthony et al. (2009); Jung et al. (2012), to the minimization of free energy Friston and Stephan (2007); Friston (2012, 2010) or the so called time-loop error in the homeokinesis approach Der (2001); Der and Liebscher (2002); Der and Martius (2012), see also Prokopenko (2008, 2009) for more details on how to organize SO. Given an objective function, the optimization process can be translated into a learning rule that is driving the SO process.

These OFs all fulfill the prerequisite of a principled approach to SO: as they are formulated in a domain invariant way, they do not determine specific directions for the autonomous development, avoiding to put in what one actually wants to get out. But this achievement creates a dilemma which is the second, more serious challenge to SO. In fact, and this seems to be the argument, if nothing is specified from outside, will SO simply make the robot an arbitrary subject that is completely unpredictable in its behaviors and thus rather a thread than a hope. The aim of this paper is to show that this attitude is not justified. Instead, we will develop an understanding of what happens if the system is self-organizing, what the role of the embodiment is and how we can find clues for predicting and shaping the behavior patterns emerging in a genuine SO scenario.

In this paper we study how a self-organizing approach to robot control can break symmetries of the robot-environment system such that structured behavior emerges. Our approach is based on a new learning rule, see Der (2013), applied to two robotic systems. By these examples, we want to make the reader aware of the phenomenon of spontaneous

symmetry breaking that is in our opinion instrumental for understanding how SO can be effective in robotic systems. We think that the robotic community so far has overlooked the importance and substance of that phenomenon. It is one aim of this paper to contribute to the dissemination of this prospective ingredient for modern embodied robotics, see also Pfeifer and Bongard (2006); Pfeifer et al. (2007).

This paper is organized as follows: The next section describes the control framework which is the basis for the definition of the unsupervised learning rule in the following section. Then we present the first robotic scenario in section “Vehicles: behavior as broken symmetry” with examples of behaviors from sparse symmetry breaking events. We formulate a “rule of thumb” for self-organizing behavior from symmetry breaking. It follows the section “The HEXAPOD” in which we study the emergent behavioral modes and control structures using a six-legged robot. Finally we conclude with a discussion. Supplementary material, especially videos, are available at <http://playfulmachines.com/ECAL2013>.

### Control Framework

Fundamental to our approach is the closed loop control setup. The controller of the robot is given by a one layer feed-forward neural network transforming sensor values  $x \in \mathbb{R}^n$  into motor values  $y \in \mathbb{R}^m$  as

$$y = K(x, C, h) = g(Cx + h) \quad (1)$$

where  $C$  and  $h$  are the parameters (synaptic strengths and bias values, respectively) and  $g_i(z) = \tanh(z_i)$  is the sigmoidal activation function. The translation between the external and the internal world can be done by a forward model predicting future sensor values on the basis of the current sensor and motor values. Here we use a linear network:

$$x_{t+1} = \phi(x_t, y_t) + \xi_{t+1} = Ay_t + Sx_t + b + \xi_{t+1}$$

where  $\xi$  is the prediction error and the parametrized function  $\phi : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  is the predictor with the parameter matrices  $A$  and  $S$ , and the parameter vector  $b$ . The forward model can be adapted on-line by a supervised gradient procedure to minimize the prediction error as

$$\Delta A = \eta \xi y^\top, \quad \Delta S = \eta \xi x^\top, \quad \Delta b = \eta \xi. \quad (2)$$

In the applications, the learning rate  $\eta$  may not be small such that the low complexity of the model is compensated by a fast adaptation process. It is one message of this paper that, due to the strong interplay with the embodiment, these very simple control structures can produce amazingly complex behaviors.

### One-Dimensional Example

Let us consider a wheeled robot on a rail with a single motor and a single wheel-counter measuring the wheel velocity.

Connecting the simple controller given by equation (1) and interpreting the motor values as target velocities we can analyze the dynamical properties of the system. Let us first consider  $h = 0$ . For  $C < 1$  there exists only one fixed point for  $x = 0$ , corresponding to the standing robot whereas for  $C > 1$  there are two fixed points one for forward and the other one for backward driving. The system is fully symmetric in this respect assuming that also the morphology is perfectly forward-backward symmetric. More formally the system is invariant against inversion of the  $x$ -axis. For  $h \neq 0$  there is an asymmetry in the bifurcation structure, which we will not discuss further, see Der and Martius (2012) for details.

At this simple example we can understand how symmetries can be broken by noise. Let the controller be given by  $C = 0, h = 0$ , such that the robot is in total rest. When we now increase  $C$  to a value larger than 1 we cross the bifurcation point and the resting state becomes unstable and the perturbations by e. g. noise decide to which fixed point the system goes.

### Unsupervised learning for self-organization

In recent work, the so called predictive information (PI) was introduced as a general objective function for SO (Ay et al., 2008, 2012; Zahedi et al., 2010). In Martius et al. (2013), a modification of the PI, the so-called time-local predictive information (TiPI) was introduced for better coping with the problem of non-stationarity in continually learning systems. By maximizing the TiPI, a general learning rule for the synaptic strengths of a neural controller network was derived. Different from infomax principles derived so far, the method interrelates the principle formulated at the level of behaviors directly down to the synaptic level.

In Der (2013), starting from the learning rule given in Martius et al. (2013), a new rule was presented. Compared to the TiPI, this new rule was shown to drive the system toward self-organization in a more sensitive way, giving rise to a rich scenario of spontaneous symmetry breaking. This was argued to open ways to new classes of self-organized behavior. A discussion will be given below.

The rule is written as (all quantities are at time  $t$ )

$$\frac{1}{\varepsilon} \Delta C_{ij} = \delta y_i \delta x_j - \gamma_i y_i x_j \quad (3)$$

$$\frac{1}{\varepsilon} \Delta h_i = -\gamma_i y_i \quad (4)$$

where  $\delta x_t$  is the prediction error based on time  $t - 1$

$$\delta x_t = x_t - \phi(x_{t-1}, y_{t-1})$$

or some other perturbation quantity<sup>1</sup>.  $\delta y_t$  is defined in terms

<sup>1</sup>The new rule is not restricted to using  $\delta x$  as the prediction error. Instead we are free to consider  $\delta x$  as any convenient change in or perturbation of the sensor dynamics. In the experiments described below we used the change of the sensor values in one time step.



of the world model as

$$\delta y_t = J^\top \delta x_{t+1} \quad (5)$$

where

$$J = \frac{\partial \phi(x, y)}{\partial y}$$

is the Jacobian matrix of the forward model expressing the sensitivity of its output on the input  $y$ . In our linear model, we simply have  $J = A$ .

Moreover,  $\gamma_i$  is a neuron specific learning rate defined as

$$\gamma_i = 2\alpha \delta y_i \delta z_i \quad (6)$$

where  $\alpha$  is an empirical quantity controlling the sensitivity with  $\alpha > 1$ , and  $\delta z = C\delta x$  is the change in the postsynaptic potential caused by  $\delta x$ .

### Discussion of the learning rule: self-induced symmetry breaking

The specific form of the learning rule allows for a very basic interpretation. Let us start with the last term  $\gamma_i y_i x_j$  contributing to  $\Delta C_{ij}$  which is easily recognized as a Hebbian term since it is the product of the input  $x_j$  into the synapse  $C_{ij}$  and the activation  $y_i$  of neuron  $i$ . As such it would strengthen all paths in the SM loop for which there is a strong output of the motor neuron combined with a strong response from the outside world as reported by the sensor value  $x_i$ . This would drive the neurons into saturation. However, with the negative sign (and  $\gamma_i > 0$ , in standard situations), the term actually is anti-Hebbian, counteracting the saturation of the neurons.

The first contribution,  $\delta y_i \delta x_j$  is Hebbian again, formulated, however, not in the activations itself but in their deviations from the predicted values as generated by the model. Given the relation between  $\delta y_t$  and  $\delta x_{t+1}$ , see equation (5),  $\Delta C_{ij}$  is strengthened if there is a strong correlation between  $\delta x_{t,j}$  and the components of  $\delta x_{t+1}$  being fed by  $\delta y_{t,i}$ . Roughly speaking, in that way the first term in the learning rule tries to increase the propagation of perturbations  $\delta x$ , driving the system towards instability. Here we can draw a parallel to homeokinetic learning (Der and Martius, 2012), where we also have two antagonistic terms which together should drive the system towards an exploratory behavior. The structure of the learning rules are similar but differ in details, as discussed below.

In the bifurcation scenario discussed above, the symmetry breaking was induced by changing the controller parameter manually. With the unsupervised learning rule, we have a self-referential system, a dynamical system that changes its parameters by itself, see also Der and Martius (2012). The decisive point in this scenario is the fact that (i) the learning rule does not introduce explicitly any violations of symmetries of the physical system it is applied to, but that (ii)

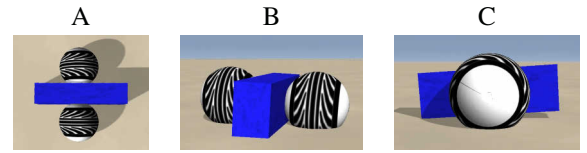


Figure 1: The TWOWHEELED as a 3D physical object. The motor values are interpreted as wheel velocities, however the simulated motors have a maximal force. The ground is elastic so that the wheels sink into the ground depending on their load. Additionally there is friction and slip. The sensors measure the actual wheel rotation velocity. (A,B) wheel size 1: the robot is lying more or less flat on the ground when driving straight. When moving in a curve, there is an inclination due to the physical forces making the effective radius of the wheels different, see the video S1. Note the wheels are about 2% off center, so the robot is not fully forward/backward symmetric. (C) Wheel size of 1.2 the body can tilt to the front and back. These 3D effects would make both odometry and the execution of motion plans very difficult as they involve the full physics of the robot.

the learning drives the physical dynamics towards instability, eventually causing a **spontaneous** breaking of existing symmetries.

To give an example: consider a hexapod robot (see e. g. Figure 7 below) where the parameters  $C_{ij}$  represent the couplings between the sensors and the motors. Intuitively the  $\delta x$  and  $\delta y$  contain some information of the current mode of behavior, that is not already modeled perfectly by the forward model. Combined in the driving term the learning rule will amplify a latent (easy to excite) mode of behavior.

### Vehicles: behavior as broken symmetry

Let us now apply the new learning rule to the specific example of a TWOWHEELED robot (Figure 1) such that the characteristic properties of the self-organization process are illustrated. For the simulations the LPZROBOTS simulator (Martius et al., 2010) was used.

#### Least biased initialization

In applications, a first point is about the choice of the initial parameters of the networks and the initial configuration of the robot. With our specific choice of the controller network, the initialization with  $C = 0$  seems most natural because this corresponds to a controller that is completely numb, i. e. deprived of any functionality. Putting additionally  $h = 0$ , we find that all motor neurons send the command  $y_i = 0$  to the motors, independently of any inputs.

Choosing the initialization in the described way has different effects on the initial pose the robot is taking. For example, in the TWOWHEELED case this means that all wheels are held at rest (velocity control). In robots with

joint-position control,  $y = 0$  means that all joints are driven towards their center position.

The combined system, comprising the physical and the synaptic dynamics, is fully deterministic. If starting in the least biased initialization the combined system may be in an unstable fixed point. We can either add small noise to the sensors for a short time interval or position the robot initially such that sufficient perturbation occur. Without further noise, the actual initial condition is fixed and the time evolution of the entire system is deterministic.

### Symmetry breaking—a rule of thumb

Before going on to present the experiments, let us formulate a simple rule of thumb on the development of the robot when starting from the least biased condition: in typical experiments we observe that the behavior of the robot can be described as being active (caused by the driving term in the learning rule ( $\delta y_i \delta x_j$ )) while conserving as much of the original symmetries of the system as possible. When only few of the symmetries are broken we call it the parsimony (or economy) of symmetry breaking. Note that symmetries involve not only the geometry of the robots body but also all the symmetries of the physical dynamics. In the two-wheel robot case the body geometry is described by left-right and forward-backward symmetries. The physical symmetries are based on the robot being an object in space and time, the physics being invariant against both translations and rotations of the frame of reference, taking however account of the physical boundaries (objects, walls, and ground). To give an example: If the robot drives in a straight line back and forth, the rotational symmetry of the space is broken, whereas the forward-backward, left-right symmetries are conserved. However, if the robot drives in a circle the rotational symmetry is conserved and the others are broken. So a 'good' behavior in the sense of parsimoniously broken symmetries would be driven in a circular pattern with both forward and backward driving.

Let us also emphasize that symmetry breaking observed in this scenario is emerging as a phenomenon “from inside” the deterministic system itself so that we may speak of a spontaneous symmetry breaking (SSB). As an additional feature, the breaking of the symmetries can largely be influenced by external impacts (physical forces in the sense of a desired mode) and/or by choosing specific sensor combinations that help to organize the symmetry breaking scenario. We will give an example with the HEXAPOD further below.

### Results

The learning starts in the least biased way, so that the symmetry breaking should follow the principle of parsimony mentioned above. In particular, the physical system is invariant against spatial transformations, i. e. translations or rotations of the spatial frame of reference. With the constraints given by the (elastic) surface, the remaining symmetry oper-

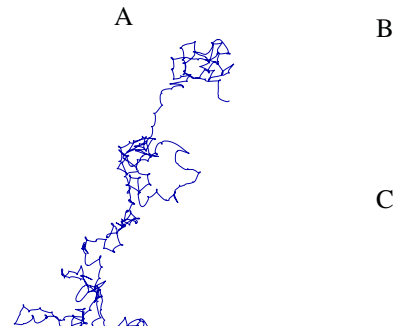


Figure 2: Deterministic trajectories of the robot in the ground plane emerging with different learning rates  $\varepsilon$ . If learning is fast ( $\varepsilon > 0.01$ ), irregular trajectories occur (A). With lower rates (here  $\varepsilon = 0.001$ ), after a transient phase of irregular motion through metastable attractors (B), the dynamics is converging toward a limit cycle behavior (C), called the master cycle below. The width of the robot is displayed by the small scale-line at the bottom. See also video S2. Parameters:  $\alpha = 3$ .

ations are rotations around the  $z$  axis and translations in the  $x - y$  plane. Remember that the learning rule gives no clue of how symmetries are to be broken.

When using the controller (equation (1) with the learning dynamics given by equations (3) and (4) (and fixed forward model with  $A = \mathbb{I}, S = 0, b = 0$ , for simplicity), we expect the robot to start moving after some time<sup>2</sup> while trying to conserve as much of the original symmetries as possible. However, when using a learning rate  $\varepsilon$  above a certain value, the robot is seen to engage in a sequence of left and right turns combined with motions back and forth along curved lines, without any regularity to be seen, see Figure 2(A). Still, note that these trajectories are fully deterministic. Nevertheless, our rule of thumb obviously is not valid in this regime as there is no visible footprint of the underlying symmetries—the invariances against rotations and translations of the physical space.

The situation changes drastically when using smaller learning rates so that the interplay between learning and physical state dynamics is given time to unfold. Figure 2(B,C) is demonstrating a typical behavior of the robot. After starting, the robot is running through a kind of metastable patterns converging after some time toward a

<sup>2</sup>When using low learning rates, this time can be very long so that we often start the robot with an initialization close to the bifurcation point, choosing  $C = c\mathbb{I}$  with  $c$  close to 1. Contrary to the HEXAPOD treated below, in the TWOWHEELED case, no substantial differences in the behaviors were observed.

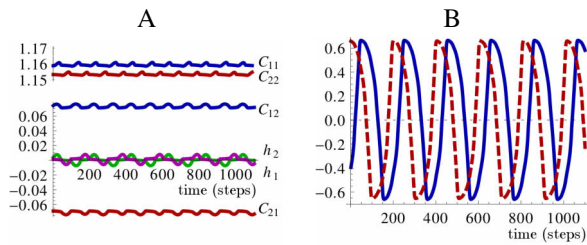


Figure 3: The circular pattern formation (Figure 2(C)) is hidden in the dynamics of the controller parameters as driven by the general learning rule. The  $C$ -matrix (A) is seen to be of a nearly perfect  $SO(2)$  structure (rotation matrix), which can be described by a single rotation angle and a scaling, so instead of four parameters there are only two required. The  $h$  dynamics is seen to be periodic with a slight bias. (B) the two sensor values (wheel velocities).

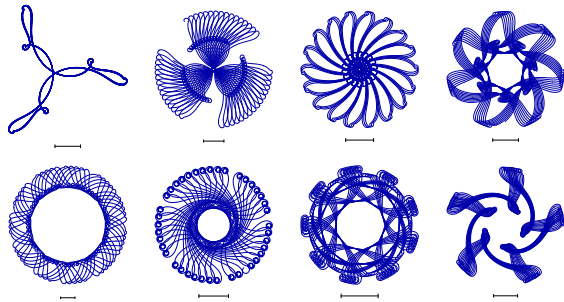


Figure 4: Patterns for frozen controller parameters occurring along the master cycle, Figure 2(C). Depicted is a selection of such patterns from parameter snapshots in one period of the state-parameter dynamics (about 200 time steps in Figure 3). If the learning is switched on again, the full dynamics is converging back to the master cycle.

large scale circular pattern (CP).

The parameters of the controller during the CP (Figure 2(C)) are not constant but run themselves through a limit cycle as displayed in Figure 3. This is an immediate consequence of the close and persistent coupling between learning and physical dynamics. What happens if we keep them fixed at any time? The answer is quite astonishing: a variety of different patterns emerges, as displayed in Figure 4. This also illustrates that the parameter dynamics within the limit cycle is actually important for the particular pattern. The former can be seen as a transient along the many stable patterns with fixed parameters. Upon switching on learning again, the system rapidly returns to the original CP (with a different spatial position). This so called pattern spin-off effect was for the first time reported in Der (2013), this paper presents additional results demonstrating the richness of that phenomenon.

At present we do not have a complete microscopic un-

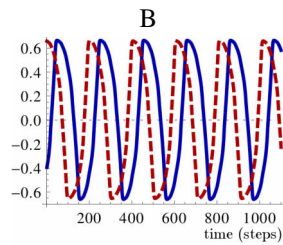


Figure 5: The emerging patterns also depend sensitively on the learning parameters. The figure shows the emerging patterns with changing  $\alpha$  parameter (from left to right) as  $\alpha = 1.0, 1.3, 1.9, 2.0$ .

A B

Figure 6: The role of embodiment. (A) Wheel size 1.1 (default 1.0). After a very irregular initial phase, the robot enters an aligned wiggly pattern, running at first to the right and then back toward the left lower corner. (B) Wheel size 1.125 leads to a circular pattern again. Parameter:  $\varepsilon = .001$ ,  $\alpha = 3$ .

derstanding of the effects. Still, at the level of phenomena, there is a number of observations. One is that the very nature of the emerging patterns depends in a most sensitive and intricate way on both the embodiment and the learning dynamics. For instance, by varying the so called sensitivity parameter  $\alpha$  (equation (6)) of the learning rule we obtain a set of quite different CPs as shown in Figure 5.

Alternatively, we may change the embodiment and obtain another class of behaviors. One option is to increase the wheel size that causes the trunk of the robot to tilt more when accelerating, see Figure 1. Two exemplifying trajectories are presented in Figure 6. For certain wheel sizes we may get also linear patterns, as they are predominant with a fully forward/backward symmetric robot. On the general level we may argue that for the linear pattern not the rotation symmetry is partially conserved, but the translational one along the line. However only a small change in the wheel size yields a CP again but with a very different fine structure.

Are we lost? Confronted with such an overwhelming variety of emerging patterns, are we faced with a robot that is behaving completely unpredictable confirming just the concerns against self-organizing robots we wanted to dispel? One answer is found by taking a look at the controller parameters. As Figure 3 shows, the controller matrix  $C$  is of a very specific structure, it is a nearly perfect (scaled) rotation matrix. Any such matrix rotates a vector by an angle and stretches it by a factor, so it is parametrized by only two

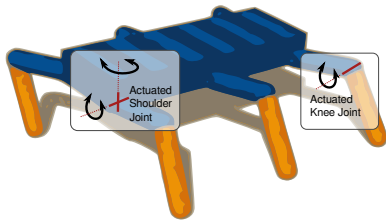


Figure 7: The HEXAPOD. 18 actuated DoF: 2 shoulder and 1 knee joint per leg. Fully forward/backward and lateral symmetric.

variables. This specific structure of the controller matrix, obtained by the learning, also seems to be responsible for the specific pattern creation effects. In principle, there are essentially three possibilities for the asymptotic system dynamics with a fixed controller matrix: fixed point, limit cycle and chaotic attractor. As we have observed in a series of experiments, the limit cycles are most likely to occur with a rotation matrix. This is intuitively understandable given that the physical space is invariant against rotations.

In a sense, this specific controller structure is like a footprint left by the symmetries of physical space, imprinted into the controller by **spontaneous** symmetry breaking, driven by the unsupervised learning procedure that does not break any symmetries by itself.

So, from looking inside, there is a coarse explanation of how the robot achieves the patterns—by discovering, so to say, the world of rotation matrices. However, the point of major interest is that the learning finds those specific structures. On a general level, an understanding may be given by the rule of thumb: a pattern in space can only emerge from breaking the spatial symmetries inherent in the physics of the robot. When trying to make this symmetry breaking as parsimonious as possible, a circle is nearly perfect: while it has broken the translational symmetry (the center is a fixed point in space), rotation symmetry (around that center) is fully conserved. Yet, because of its fine structure, the actual patterns emerging in the learning scenario are not circles but CPs. Nevertheless, they are still invariant against rotations about a definite angle, see in particular the patterns of Figure 4 and Figure 5. This may be seen as a noteworthy parallel to the hexagonal patterns known from many phenomena in nature. So, the observed patterns apparently are the ones with a high degree of preserving the spatial symmetries of the physical system.

### The HEXAPOD

Let us now follow the trace of symmetry breaking with a high-dimensional six-legged robot: the HEXAPOD, see Figure 7. We choose this robot because it will be seen to reveal symmetry breaking phenomena in a particularly clear way. The robot has six legs, each one with three degrees of freedom (DoF). Each of the 18 joints is actuated by a servo mo-

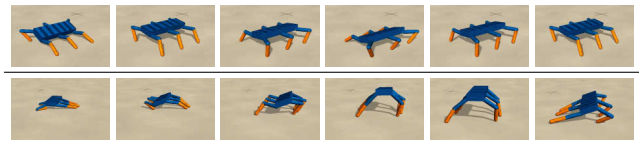


Figure 8: Initially, after about 20 min the robot develops a swaying motion pattern (top row), as if it is very actively trying to move the legs in a coherent way while keeping ground contact. 50 min later a raising behavior develops where the trunk is repeatedly being lifted from the ground.

tor and contains a sensor that is measuring the actual joint angle. The effective torques acting on the joint axes are determined by a PID controller with a limited force. To enable a body feeling (some useful feedback from the interaction), this force limit is proportional to the deviation from the set point, such that there is an elastic reaction to external forces, similar to a system controlled by muscles.

In a typical experiment, the HEXAPOD is falling down from a starting position a little above the ground. With the least biased initialization the motor values are zero ( $y = 0$ ) so that all joints are in their center positions. When hitting the ground, the robot gets into a damped vertical oscillation due to the elasticity of the joint-motor system. This is sufficient for providing an initial perturbation  $\delta x$  that is further amplified by the learning dynamics.

What can we expect to happen? Depending on the concrete situation (e.g. particular  $\varepsilon$ ) different behaviors may emerge. In most cases the robot starts with a swaying motion pattern, see Figure 8 and video S3. We may claim again, that this is in agreement with our rule of thumb since in this motion the joint angles are changing with a pretty high degree of coherence as allowed by the physical constraints enforced by the ground contacts.

More interesting behaviors emerge after some time, for instance a raising behavior, see Figure 8. The entire development can be followed in short pieces in the videos S3-S6. There is another surprise—when looking at the parameters of the controller. In the *TWO WHEELED* case the *C*-matrix developed into a rotation matrix. Of course, we can not expect such a clear result in the case of our HEXAPOD because of the much higher dimensionality of the physical space and the interaction with the ground.

Yet, as Figure 9(A) shows, the emerging sensor-to-motor coupling matrix is highly structured, reflecting the original symmetries to a high degree. Both the shoulders vertical direction and the knees are seen to follow essentially the same strategy for moving the body. This is in agreement with our rule of thumb since this collective strategy allows the body to be moving, but with a maximum degree of coherence between the individual constituents of the body. Moreover, the coupling matrix reveals the whole-body nature of the behavior—the control for each body part is generated by



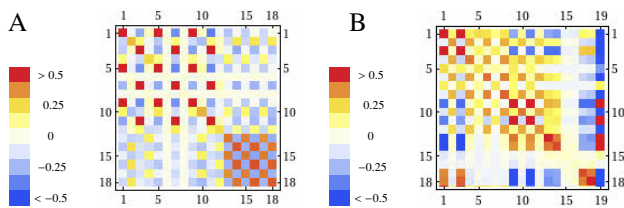


Figure 9: The parameters of the controller ( $C$ -matrix) for two scenarios. Element  $C_{ij}$  represents the coupling from sensor  $j$  to motor  $i$ . Indices: 1-12: shoulders (vertical/horizontal), 13-18: knees. (A) Swaying motion, Figure 8. (B) Seesaw motion with velocity sensor (index 19), see Figure 10. The difference between the swaying and seesaw behavior are clearly visible in the structure of the matrices. While in swaying all legs follow the same strategy, the antiphase nature of the seesaw behavior is reflected in the different sign distribution of the matrix elements.

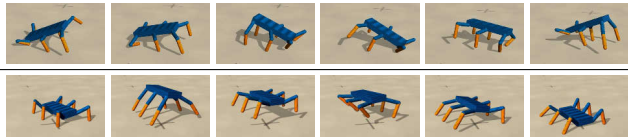


Figure 10: Seesaw motion pattern with forward/backward speed sensor (top row), see video S7. Jumping motion pattern emerging with vertical speed sensor (bottom row), see video S8. Note the robot is in the air in the frames 2-4.

combining both excitatory and inhibitory signals from the sensors of all joints in a systematic manner.

### The formative power of exteroceptive sensors

Up to now we were using only proprioceptive sensors so that the orientation of the robot in space can only be measured very indirectly, e.g. by additional load to the joint motors due to gravity forces. By including exteroceptive sensors, the development of the modes can be influenced and driven into desired directions. Adding a sensor measuring the forward velocity of the robot generates a seesaw motion. In contrast, a sensor measuring the vertical velocity of the robot leads to a pronounced jumping behavior, see Figure 10. Also here, we find highly structured controller matrices, see Figure 9(B) for the seesaw case. Note the strong coupling of the exteroceptive sensor to the motor neurons showing the functional role of that sensor. It distinguishes forward and backward motion and thus this symmetry is lost, so that in the learning process behaviors with broken forward-backward symmetry are favored.

## Discussion

This paper tries to answer essentially two questions. The first question is about how to organize self-organization, in other words, how can we find intrinsic mechanisms that

make a system able to self-organize. The answer was given by the unsupervised learning rule (ULR), see equations (3) and (4), which fulfills the main criterion for a genuine SO: it is universal in the sense that the only necessary information about the system is given by the number of sensors and that of motor neurons, any further information being acquired by the co-learning self-model in a bootstrapping process.

The second question we want to answer in this paper is suggested by exactly that bootstrapping scenario: with nothing specified from outside, what can we expect the learning system to do. What will the emerging behaviors look like and what will the relation to the embodiment of the robot be? How and to which extent are the emerging behaviors determined by the embodiment; and can we find systematic criteria for those behaviors?

Several answers could be given by looking into the role of the underlying symmetries of the system in space and time which induces, given the constraints, corresponding symmetries in the physical system. The point then is that, while driving the system towards instability, the ULR is preserving these symmetries. As a result, the evolution of the system in the learning process is realized by a sequence of spontaneous symmetry breaking steps, following—similar to what we know from nature—a kind of parsimony principle. This leads to our rule of thumb: the emerging behaviors in physical systems (robots) driven by our ULR are qualified by a high activity while preserving as much of the underlying symmetries as possible.

This rule brings the embodiment to the foreground. The symmetries are embodiment specific and, moreover, breaking the symmetries is a process that is related to the very physics of the system. This was demonstrated by a number of examples. The first and probably the most surprising one was given by the TWO WHEELED robot. Controlled by two neurons with a fast synaptic dynamics given by the ULR, the system in many cases was converging towards a limit cycle behavior with the trajectories of the robot forming nearly perfect geometric patterns. The emerging geometric patterns were seen to depend on the embodiment (like the wheel size) in a very intricate and sensitive way. Interestingly, the limit cycle acts as a pattern factory: the parameters occurring along the limit cycle produce a great variety of spin-off patterns. While this effect has already been reported in Der (2013), this paper presents further results and gives additional insights into this interesting effect.

Continuing the work started in Der (2013), similar effects of symmetry breaking were obtained in the example of the HEXAPOD. We observed the excitation of body related, high activity modes with a high degree of coherence between the body parts. These modes were argued to be in nice agreement with our rule of thumb: emerging behaviors are qualified by high activity while preserving the underlying symmetries of the system as far as possible (the principle of parsimony in spontaneous symmetry breaking). In future work

we will be looking for a parallel of the pattern spin-off effect, hoping to thereby uncover a kind of pattern factory for these more complex systems, too.

These results are a step forward as compared to the state of the art. Previous work in self-organizing robot behavior was either restricted to small, easy to analyze systems or produced—like with the principle of homeokinesis—behaviors which looked interesting and were often completely surprising (Der and Martius, 2012), as it should be. However, by the same argument, it was often not clear what the robot is actually doing. With the new learning rule and the concept of behaviors as broken symmetries, this is now (a little) different. The essential difference between homeokinetic learning and the ULR is the dynamics with the least biased initialization (“do nothing” region with all synapses zero). While the time-loop error of homeokinesis has a pole there, the infomax based objectives are smooth in that region. It is basically this smoothness that makes the learning to integrate the responses of the system dynamics in a sensitive way. As compared to the TiPI (Martius et al., 2013), the learning dynamics used here is even smoother and even more concentrated on system responses which explains the prevalence of spontaneous symmetry breaking effects. At a more formal level, we see the difference also in the drift of the local Lyapunov exponents: while homeokinesis drives small exponents stronger than larger ones, the situation is inverted in the present learning dynamics. Given the formative interplay between state and learning dynamics, this has important consequences for the emergence scenario of the behaviors.

The principles and examples given in this paper—in particular the emergence of coherent modes, the TWO-WHEELED as a pattern factory and the various modes realized by the HEXAPOD—may help us to better understand and exploit the synergy between embodiment and SO of autonomous robots.

## Acknowledgments

The authors gratefully acknowledge the great hospitality in the group of Nihat Ay. We thank for many helpful and clarifying discussions with Nihat Ay, Nils Bertschinger, Eckehard Olbrich, and Keyan Zahedi. GM was supported by a grant of the DFG (SPP 1527).

## References

- Anthony, T., Polani, D., and Nehaniv, C. L. (2009). Impoverished empowerment: ‘meaningful’ action sequence generation through bandwidth limitation. In *ECAL*, pages 294–301.
- Ay, N., Bernigau, H., Der, R., and Prokopenko, M. (2012). Information driven self-organization: The dynamical systems approach to autonomous robot behavior. *Theory Biosci.*, 131(3):161–179.
- Ay, N., Bertschinger, N., Der, R., Güttler, F., and Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *The European Physical Journal B*, 63(3):329–339.
- Der, R. (2001). Self-organized acquisition of situated behaviors. *Theory Biosci.*, 120:179–187.
- Der, R. (2013). On the role of embodiment for self-organizing robots: behavior as broken symmetry. In Prokopenko, M., editor, *Guided Self-Organization: Inception*. Springer, to appear.
- Der, R., Güttler, F., and Ay, N. (2008). Predictive information and emergent cooperativity in a chain of mobile robots. In *Artificial Life XI*. MIT Press.
- Der, R. and Liebscher, R. (2002). True autonomy from self-organized adaptivity. In *Proc. Workshop Biologically Inspired Robotics*, Bristol.
- Der, R. and Martius, G. (2012). *The Playful Machine - Theoretical Foundation and Practical Realization of Self-Organizing Robots*. Springer.
- Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature reviews. Neuroscience*, 11(2):127–138.
- Friston, K. J. (2012). A free energy principle for biological systems. *Entropy*, 14(11):2100–2121.
- Friston, K. J. and Stephan, K. E. (2007). Free-energy and the brain. *Synthese*, 159(3):417–458.
- Jung, T., Polani, D., and Stone, P. (2012). Empowerment for continuous agent-environment systems. *CoRR*, abs/1201.6583.
- Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005). Empowerment: a universal agent-centric measure of control. In *Congress on Evolutionary Computation*, pages 128–135.
- Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2007). Representations of space and time in the maximization of information flow in the perception-action loop. *Neural Computation*, 19:2387–2432.
- Martius, G., Der, R., and Ay, N. (2013). Information driven self-organization of complex robotic behaviors. preprint, submitted to PLoS ONE.
- Martius, G., Hesse, F., Güttler, F., and Der, R. (2010). LPZROBOTS: A free and powerful robot simulator. [robot.informatik.uni-leipzig.de/software](http://robot.informatik.uni-leipzig.de/software).
- Pfeifer, R. and Bongard, J. C. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press, Cambridge, MA.
- Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science*, 318:1088–1093.
- Prokopenko, M., editor (2008). *Foundations and Formalizations of Self-organization*. Springer.
- Prokopenko, M. (2009). Information and self-organization: A macroscopic approach to complex systems. *Artificial Life*, 15(3):377–383.
- Zahedi, K., Ay, N., and Der, R. (2010). Higher coordination with less control – A result of information maximization in the sensorimotor loop. *Adaptive Behavior*, 18(3-4):338–355.

# Social Inhibition Manages Division of Labour in Artificial Swarm Systems

Payam Zahadat<sup>1</sup>, Karl Crailsheim<sup>1</sup> and Thomas Schmickl<sup>1</sup>

<sup>1</sup>Artificial Life Lab of the Department of Zoology, Universitätsplatz 2,  
Karl-Franzens University Graz, 8010 Graz, Austria  
payam.zahadat@uni-graz.at

## Abstract

This paper presents a novel and bio-inspired algorithm for distributed division of labour in swarms of artificial agents (e.g., autonomous underwater vehicles). The algorithm is inspired by division of labour via local interactions in social insects. The algorithm is successfully implemented in virtual agents and simulated robot swarms and demonstrates a high adaptivity in response to changes in the workforce and task demands in the swarm level as well as a high specialization to tasks in the agents level.

## Introduction

In the field of swarm intelligence and collective robotics, social insects are promising sources of inspiration due to their capabilities in self-organization and self-regulation in the colonies. Division of labour is one of the prominent characteristics of social insects such as honey bees (Seeley (1982); Huang and Robinson (1992)) and ants (Hölldobler and Wilson (2008); Julian and Cahan (1999)). The insect colonies maintain plasticity that adapts the division of labour to the status of the colony and, in parallel, also to environmental situations, e.g. to the number of workers or task-demands. The flexibility and quick response to the changes in the status of the colony and environment in one hand and specialization of workers for the tasks in the other hand are interesting properties of the methods driving behaviour in the colonies.

Several models have been proposed to explain the mechanisms of division of labour in social insects, e.g. foraging-for-work (Tofts (1993)), response-threshold reinforcement (Bonabeau et al. (1997); Theraulaz et al. (1998)), and common-stomach models (Karsai and Schmickl (2011)) (for a good review of such models see Beshers and Fewell (2001)). These models have been also used as inspiration for distributed algorithms in application areas such as swarm robotics. For example, in Gross et al. (2008) and Labella et al. (2006), models of ants-foraging behaviour have been implemented for maintaining division of labour in a group of robots performing an object retrieval task or response-threshold reinforcement model (Bonabeau et al.

(1997); Theraulaz et al. (1998)) that is inspired by wasps is applied for swarms of robots (e.g. in White and Helferty (2005); Yang et al. (2009)), or in Schmickl et al. (2007) a *trophallaxis-inspired* strategy which is inspired by food exchange of honeybees and ants is applied to a simulated robot swarm.

In this paper, we are interested in a mechanism of division of labour that is inspired by behaviours of honey bees (Huang and Robinson (1992, 1999)). A honey bee undertakes different tasks during its life-time in a process of behavioural development. In earlier weeks of its adult life, a honey bee performs nursing, then it performs other tasks inside the hive, and only in its final weeks it leaves the hive for foraging (Johnson (2010)). This behavioural development can be delayed, accelerated, or even reversed in response to changes in colony or environmental conditions. *Social inhibition* is proposed (Huang and Robinson (1992)) as a conceptual method for maintaining this adaptive behaviour of the colony. In this method, tasks are considered in an ordered sequence. The behavioural development of an individual that determines when the worker switches to the neighbouring task in the sequence is regulated via local interactions with other individuals.

In this paper, a distributed algorithm of division of labour based on local communication is inspired by social inhibition. The algorithm considers the spatiality of the task-regions that restrict the possible local interactions between the individuals as it is more realistic regarding many application areas and also the biological system. For example, in a honey bee colony, the workers of the tasks which are early in the sequence stay inside the hive while the workers of the tasks later in the sequence work outside the hive. The very early workers do not have much contact with the out-of-the-hive workers. In other words, the interactions are restricted to some extent to the individuals of the tasks which are next to each other in the sequence.

Despite the biological source of inspiration of the proposed algorithm, we aim for applications in swarm robotics. In particular, where there is a spatial clustering of the robots based on their tasks that limits the local communications

to the robots of the same or neighbouring tasks. The proposed algorithm is simple and easy-to-implement. It is implemented in swarms of virtual agents as well as swarms of simulated robots that perform several tasks. The behaviour of the swarm in response to the changes in the number of agents in each task and the task-demands are investigated representing adaptability that is achieved by the algorithm.

### Social Inhibition

In honey bee colonies, division of labour is mainly based on the age of the workers. This mechanism is called *temporal polyethism*. In temporal polyethism, there is a correlation between the age of the workers and the tasks they perform; e.g. older workers perform tasks outside of the hive and younger workers perform tasks within the hive (Wilson (1971); Robinson (1992)). The behavioural development of the bees is associated with their physiological development such that the physiological age of a bee indicates the main task that it performs (Winston (1987); Beshers et al. (1999)).

As it is shown in different studies (e.g. Huang and Robinson (1996)), honey bee colonies are flexible to changes in age distribution of the colony and task demands. For example, in a colony of young honey bees, the age in which a bee starts foraging (an outside task) is lower than in a normal colony. It means the behavioural development in such colony is accelerated. On the other hand, presence of older bees delays or inhibits the development of physiological age of other bees in the colony. Another example is the behaviour of the colonies when the hive workers are removed. In this case, the development of physiological age decreases and inverts resulting in transformation of out-of-hive workers into inside-the-hive workers.

Huang and Robinson (1992) proposed that worker-worker interactions drive mechanisms of hormonal regulation in bees resulting in a social inhibition that explains temporal polyethism and adaptability of the colony to different age distributions. The concept is then used in other researches toward developing models of social inhibition, e.g., Beshers (2001); Naug and Gadagkar (1999); Gadagkar (2001). In this work for some reasons Naug and Gadagkar (1999); Gadagkar (2001) are not used as a source of inspiration.

### Previously Suggested Algorithm: Evolution Maps

One of the models of social inhibition following Huang and Robinson (1992) is the model of *evolution maps* proposed by Beshers (2001). In this model, a “map” which is a set of curves describing changes in the physiological age of the individual is introduced. A state variable  $x$  represents the physiological age of every individual. In addition to that, the individuals also contain an auxiliary variable  $y$ . In every time-step (day for bees), an individual has a number of interactions with others.  $y$  is a weighted average of  $x$  values of all the interacted individuals. The weights are set based on the task the individual is performing. Every curve of the

“map” describes changes of  $x$  based on its current value and the value of  $y$ .

In the reported work (Beshers (2001)) two tasks are implemented. A threshold is set to indicate which task is chosen by the individual based on its  $x$  value. The threshold is augmented with higher and lower margins providing more stability for the system. The curves are derived based on experimental data from real bees.

Although the model might be extendible to more tasks, global-range interactions (interactions between individuals irrespective to the task they are performing) would be a necessary condition for the stability of the model. The reason is that if the interactions are restricted, e.g. to the neighbouring tasks in the task-sequence, the  $y$  value is no longer an estimation of the  $x$  in the whole system and will have instantaneous changes when an individual switches between two tasks leading to endless back and forth switchings.

Apart from the complexity of generating a proper map, the global-range interaction is usually not the case in both insects and robotic tasks, since workers of different tasks are usually separated spatially and interactions are restricted to individuals of the same or neighbouring tasks.

### New Proposed Algorithm

One of the properties that we are interested in them are the ability of the decentralized algorithm to divide the swarm into groups relative to the task-demands while the system is flexible to changes in the demands and workforce. In addition, the number of switchings between different tasks should be limited due to practical costs (e.g., a robot may need to spend some energy to change its working area in order to perform a different task). Therefore, specialization of the individuals is also of our interest.

In the proposed model, every individual contains a state variable  $x$  as its physiological age. This variable is restricted to a defined range of  $(x_{min}, x_{max})$ . There is a number of tasks with their associated demands. The tasks are ordered in a sequence such that an individual can only switch to the previous or the next task in the sequence.

An individual chooses a task based on its  $x$  and a set of defined thresholds that separate the tasks (see Figure 1). The thresholds are used together with lower and upper margins. For an individual that is performing  $task_i$ , in order to switch to  $task_{i+1}$ , its  $x$  value should exceed  $th_{i:i+1} + upper\_margin$ . For an individual that is performing  $task_i$ , in order to switch to  $task_{i-1}$ , its  $x$  value should become lower than  $th_{i-1:i} - lower\_margin$ . The lower and upper margins prevent the individuals from instant back and forth switches between two consecutive tasks due to noisy changes in  $x$ .

The main idea of the algorithm is to spread the  $x$  values of the whole swarm uniformly over the range of  $(x_{min}, x_{max})$ . With such a uniform distribution of  $x$  throughout the swarm and setting the thresholds such that the range is split into segments relative to the task-demands, the required number



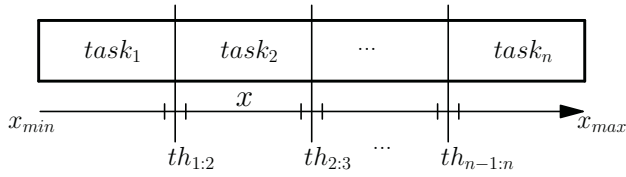


Figure 1:  $x$  changes with small steps in a range of  $(min_x, max_x)$ . A task is assigned to the individual when its  $x$  variable passes the respective thresholds.

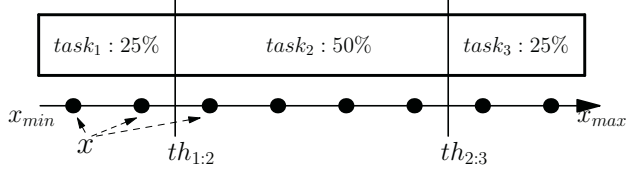


Figure 2: An example uniform distribution of  $x$  and proper thresholds that split the range relative to the task-demands. In this example, eight agents are divided into three tasks. The relative demands for the three tasks are 25%, 50%, and 25% respectively.

of agents will be assigned to each task (see Figure 2).

Regulation of  $x$  values in the swarm occurs through local interactions between the individuals. Every individual contains two variables that keep track of its experience in the swarm. In an ideal case, these two variables stand for the closest higher and lower  $x$  values belonged to other individuals in the swarm. In practice, these variables have to be estimated during interactions. Let these variables be  $x^{low}$  and  $x^{high}$ . In the ideal case their variables are as follows:

$$\begin{aligned} x^{low} &= \arg \min_{x_i} (|x - x_i|), \text{ where } x_i < x \\ x^{high} &= \arg \min_{x_i} (|x - x_i|), \text{ where } x_i > x \end{aligned} \quad (1)$$

The value of  $x$  is updated for each agent in the direction towards the average of its  $x^{low}$  and  $x^{high}$ , as follows:

$$x = \begin{cases} x + \delta & \text{if } x - x^{low} < x^{high} - x \\ x - \delta & \text{if } x - x^{low} > x^{high} - x \\ x \pm X & \text{otherwise} \end{cases} \quad (2)$$

where  $\delta$  is *step-size* which is a constant parameter with a small value in terms of the size of segments for every task. In the current implementation  $X \sim \delta \times U(0, 1)$ .

Since there is no global information about the  $x$  values in the swarm, agents update their  $x^{low}$  and  $x^{high}$  values gradually during time and in every interaction with another agent.

Say agent  $i$  interacts with agent  $j$ .  $x^{low}$  and  $x^{high}$  are updated as follows:

$$x_i^{low} = \begin{cases} x_j & \text{if } x_i^{low} < x_j < x_i \\ x_i^{low} & \text{otherwise} \end{cases} \quad (3)$$

$$x_i^{high} = \begin{cases} x_j & \text{if } x_i < x_j < x_i^{high} \\ x_i^{high} & \text{otherwise} \end{cases} \quad (4)$$

$x^{low}$  and  $x^{high}$  also slowly drift away from  $x$  in every time-step in order to be adaptable to changes in other agents'  $x$  values as well as the environment:

$$\begin{aligned} x_i^{low} &= x_i^{low} - \varphi \\ x_i^{high} &= x_i^{high} + \varphi \end{aligned} \quad (5)$$

where  $\varphi$  is a value smaller than  $\delta$  in Eq. 2.

After every update of  $x$ , an individual considers switching to the previous or next tasks in the task sequence. For an individual with  $task_i$ , *new\_task* is chosen as follows:

$$new\_task = \begin{cases} task_{i+1} & \text{if } x > th_{i:i+1} + l_u \\ task_{i-1} & \text{if } x < th_{i-1:i} - l_b \\ task_i & \text{otherwise} \end{cases} \quad (6)$$

where  $th_{i-1:i}$  and  $th_{i:i+1}$  represent the threshold values between  $task_{i-1}$  and  $task_i$ , and  $task_i$  and  $task_{i+1}$  respectively.  $l_u$  and  $l_b$  are the upper and lower margins for the thresholds.

**The Algorithm** The following actions are performed by any agent  $i$  in every time-step of running:

1. update  $x^{low}$  and  $x^{high}$  using Eq. 5.
2. if there is an interaction with agent  $j$ :
  - (a) update  $x^{low}$  and  $x^{high}$  using Eq. 3 and Eq. 4.
  - (b) update  $x$  using Eq. 2.
  - (c) update the assigned task using Eq. 6.

## Experiments

A number of experiments are performed in order to investigate the performance of the proposed algorithm, its adaptivity to changes, and specialization of the agents to the tasks.

In the first set of experiments a swarm of virtual agents is simulated and the interaction between the agents performing the same task or in the adjacent tasks occur based on defined probabilities. The sensitivity of the algorithm to the chosen values for the step-size  $\delta$  is also investigated.

In the second experiment, a simulated swarm of moving agents (robots) is investigated and the interactions are based on the location of the agents in the arena in every time-step of the simulation.

### Virtual swarm experiments

The algorithm is first tested in a number of virtual swarms of agents.

In all of the following experiments a sequence of five different tasks is considered. Every experiment is repeated for

25 independent runs. Every run is simulated for 50000 time-steps while each agent runs the proposed algorithm. Interactions are possible between the agents of the same task or neighbouring tasks. In every time-step of the simulation, 30 interactions are sampled from a uniform distribution over all the possible interactions. All the agents are initialized by  $x_{min}$  for the state variable  $x$ . Experimental settings of the algorithm are represented in Table 1.

**Investigation of the algorithm with fixed settings** In the first experiment the demands for the five tasks are equal. The progress of the number of robots in each task over time is represented in Figure 3 (left). Since the agents start with the same initial value for  $x$  ( $x_{min}$ ), they all start with *task1*. By occurring interactions during time the  $x$  values are spread in the range and the swarm is split for performing different tasks.

In the second experiment the demands for the five tasks are 10%, 40%, 10%, 30%, 10% respectively. Figure 3 (right) represents the results.

All the runs reached the stable desired status for both experiments. The experiments were also repeated while the possible interactions between the agents of neighbouring tasks were limited to a fraction of the agents in each task instead of the whole agents and similar results were achieved (data not shown).

#### Investigation of adaptivity to changes in task-demands

In order to investigate adaptability of the swarm to changes in the task-demands, another experiment is performed starting with equal demands for every task. The task demands are then changed at time-steps 20000 and 40000.

Figure 4 (left) represents the results for this experiment. As the figure demonstrates, the swarm immediately reacts to the changes in the demands. The reason is that the  $x$  values of the swarm are spread almost uniformly in the range (in the ideal situation they are spread uniformly). By changing the task-demand, the thresholds over the range of  $x$  are changed such that the range is split relative to the new settings. Therefore, proper fractions of agents are reassigned for different tasks while the  $x$  values do not need to change.

#### Investigation of adaptivity to changes in work-force

In the next experiment the adaptability of the swarm to the changes in the number of agents presented in each task is investigated. In order to do that, the experiment starts with a swarm of 30 agents. In time-step 15000, 20 agents including all the agents in *task2* and *task3* are removed from the swarm. Later on in time-step 30000, 10 more agents are introduced in the swarm in the *task1*.

Figure 4 (right) represents the behaviour of the swarm in response to these changes. As the figure demonstrates, the swarm reacts to these changes by switching the tasks of the proper number of agents. The mechanism behind this reaction is as follows: When a number of agents are removed

from the system (or new agents are introduced), the uniformity of the distribution of  $x$  in the swarm is violated. The agents with  $x$  values close to the low-density (or high density) regions in the distribution react to this situation by shifting their  $x$  value towards the region (or in opposite direction). The process continues until the distribution becomes uniform again.

**Investigation of specialization** In order to investigate specialization of the agents for the tasks in the swarm, the number of non-necessary task-switchings of every agent is evaluated during the run-time. The settings are the same as the first experiment: fixed equal demands for all the five tasks. Since all the agents start with *task1* due to initial value for  $x$ , a certain number of switchings from a task to the next one is necessary for a certain number of agents. Moreover, any switching to a previous task (switch-back) is not desirable. Figure 5 represents the frequency of switch-backs during 50000 time-steps for all runs and agents for different values of step-size ( $\delta$ ).

**Investigation of the effects of step-size** The step-size  $\delta$  in Eq. 2 is a predefined parameter in the current implementation of the algorithm. Therefore, the sensitivity of the algorithm to this parameter is investigated by repeating the first experiment with different values for  $\delta$ . Figure 6 represents a comparison for different values. The main figure demonstrates the median error of the task allocation (number of agents in wrong tasks) over time. The inline figure compares the time required for reaching a swarm-state stabilized in maximum of 5% error.

As it is represented in Figure 6, for very small values of  $\delta$ , error decreases slowly. It is more quick for middle values. For high values of  $\delta$ , the error decreases quickly regarding the main figure, but regarding the inline figure the time to reach the stable status with maximum of 5% error is high. In addition, the sizes of quartiles are big indicating that in different runs different values are calculated for the time-to-reach. The instability of the high values is also visible in Figure 5 that represents the frequency of switch-backs for different  $\delta$  for all the runs during 50000 time-steps. In this figure, for  $\delta = 0.01$  there was not a single switch-back in all the runs. As  $\delta$  increases, the frequency of switch-backs also increases. In short, if the step-size ( $\delta$ ) is too small, the system is less reactive, and convergence takes longer. But if the value is too big, the system gets instable and results differ from case to case.

#### Simulated robot experiment

In this experiment a simulated robot swarm running the proposed algorithm is investigated for its behaviour and adaptability to the changes in the workforce and task demands.

A square arena is set up with a light source located in one side (see Figure 7). A swarm of robots is supposed to be split

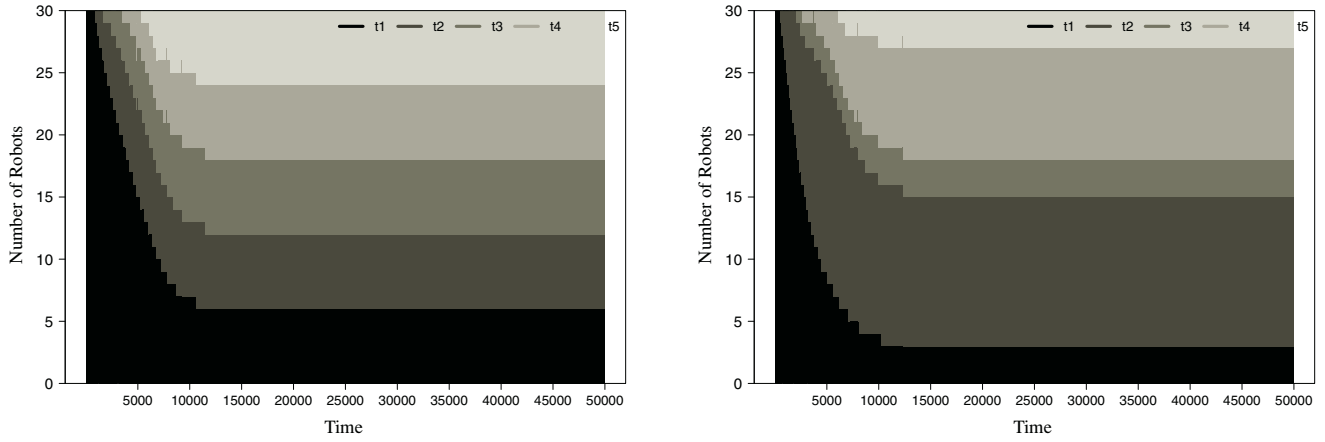


Figure 3: Medians of the number of agents in each task over time for 25 independent runs. All of the runs reached the stable solution for both experiments. Tasks are represented by  $t_1$ ,  $t_2$ ,  $t_3$ ,  $t_4$ ,  $t_5$ . The left figure represents five tasks with fixed equal demands. The right figure represents five tasks with fixed demands of 10%, 40%, 10%, 30%, 10% respectively. Number of agents in both experiments are 30.

Time	0	20000	40000
$th_{1:2}$	20	10	10
$th_{2:3}$	40	50	20
$th_{3:4}$	60	60	30
$th_{4:5}$	80	90	60
Swarm Size	30	30	30

Time	0	15000	30000
$th_{1:2}$	10	10	10
$th_{2:3}$	50	50	50
$th_{3:4}$	60	60	60
$th_{4:5}$	90	90	90
Swarm Size	30	10	20

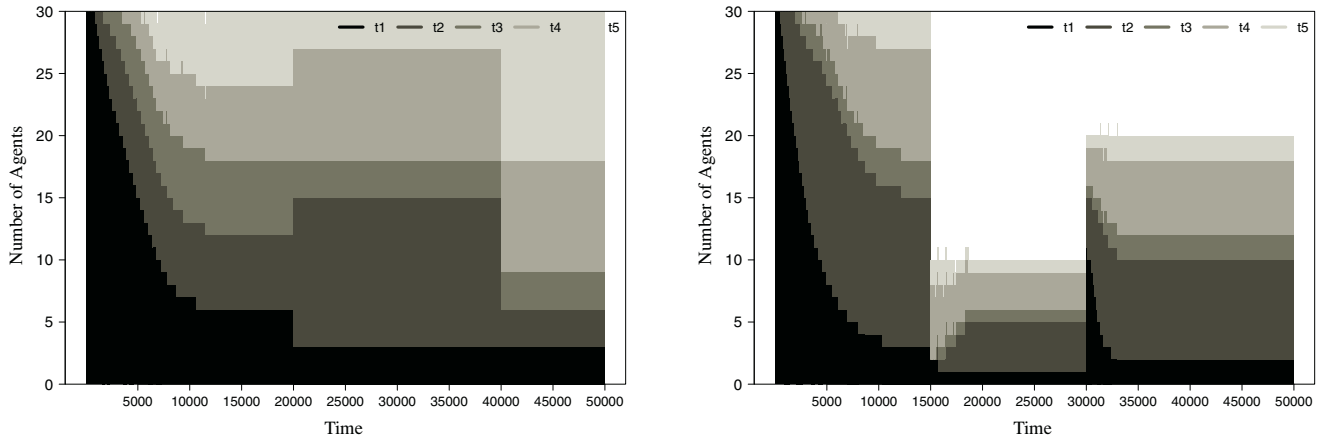


Figure 4: Medians of the number of robots in each task in 25 runs. All of the 25 runs reached the solution for both experiments. In both experiments the swarm starts with 30 agents. The left figure represents changes in the demands in time-steps 20000 and 40000. The right figure represents changes in the number of robots. In time-step 15000, 20 agents including all the agents in second and third task and randomly chosen agents are removed from the system. In time-step 30000, 10 agents are added to the system in the first task.

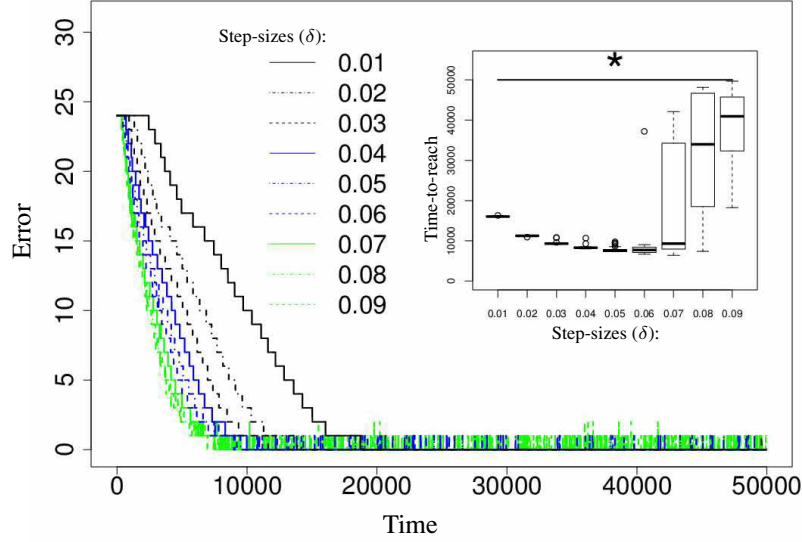


Figure 6: Comparison of error over time and also time-to-reach the stable status for nine different values of step-size  $\delta$  based on 25 independent runs. Main figure represents the median of the number of agents in wrong tasks (errors) over time for 25 runs. Inline figure represents the boxplots of the time to reach a swarm state that is stable with maximum one error out of 30 agents (more than 95% correct task-assignment) (\*: $p < 0.01$  for all pairs of  $\delta$  except (0.01,0.07) where  $p < 0.25$ ; Wilcoxon signed-rank test, unpaired data, "two.sided"-hypothesis). Box-plots indicate median and quartiles, whiskers indicate minimum and maximum, circles indicate outliers.

Table 1: Experimental settings for the proposed social inhibition algorithm

	virtual scenarios	robot scenario
$x_{min}$	0	0
$x_{max}$	100	100
$\delta$	0.03	0.1
$\varphi$	$\delta/30$	$\delta/3$
$l_u, l_b$	1	1

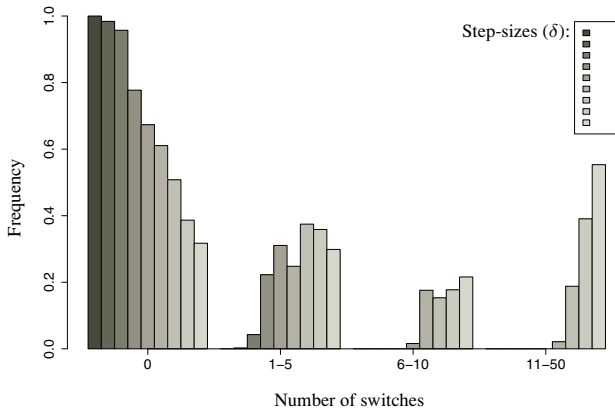


Figure 5: Frequency of number of switch-backs in 50000 time-steps for all the 30 agents and 25 runs. The figure represents the frequency for different values of step-size  $\delta$ .

up into three regions of the arena in order to cover the arena with different densities. Every robot has a luminance sensor perceiving the brightness of the light sensor. The three regions are located in different distances from the light source and the robots identify them based on their brightness. Each robot is able to rotate or move forward. When a robot decides to switch the task, it moves uphill or downhill the luminance gradient until it reaches the appropriate working region. Robots move randomly in their working region and do not leave it unless they decide to switch the task. A robot can interact with another robot which is located in its communication range by exchanging the  $x$  values. The arena is of size  $32 \times 32$  and the communication-range is five times bigger than the robots diameter.

Every robot  $i$  in the simulation performs the following:

- If the robot is in the region of its assigned task:



- Perform a random walk inside the task-region.
- Otherwise (the robot is out of its task-region):
  - turn towards the proper task-region based on the brightness gradient and step forward.
- regulate the state variables:
  1. update  $x^{low}$  and  $x^{high}$  using Eq. 5.
  2. if there is a robot  $j$  in the communication range:
    - (a) update  $x^{low}$  and  $x^{high}$  using Eq. 3 and Eq. 4.
    - (b) update  $x$  using Eq. 2.
    - (c) update the assigned task using Eq. 6.

At the beginning of the experiments, the number of robots is 16 and the task-demands are 25%, 50%, and 25% respectively which means 4, 8, and 4 robots are desired for each task-region. At time-step 20000, all the 8 robots in *task2* are removed from the arena but the proportional demands are fixed. At time-step 40000, the demands are changed to 25%, 25%, and 50% respectively. The experiment is repeated for 25 independent runs.

Figure 8 represents the progression of the number of robots in each task over time. As it is demonstrated in the figure, the swarm reacts properly to the changes in the number of robots by switching a proper number of robots from *task1* and *task3* into *task2*. The system also quickly responds to the changes in the task-demands by switching two robots from *task2* to *task3*.

Figure 9 represents the frequency of switch-backs during the first 10000 time-steps of all the runs. The figure represents that about 0.47% of the robots have not a single switch-back during the whole evaluation-time and very few robots had more than 10 switch-backs representing specialization for the robots in the tasks they perform.

## Conclusion

This paper introduces a novel decentralized, self-organized and self-regulated division of labour in artificial swarms inspired by temporal polyethism in honey bees. The algorithm is based on local communication while the communications need to be possible only between the agents that perform the same task or neighbouring tasks. The logic behind the algorithm is simple and it is easy to implement while the interesting properties are maintained for the swarm. Experiments investigating the behaviour of the swarm in response to changes in the swarm members or task-demands represents that the algorithm provides a high adaptivity for the swarm. In addition, it is demonstrated that the agents are specialized in the tasks and unnecessary switchings between the tasks are limited. The sensitivity of the system to a pre-defined parameter of the algorithm (step-size) is also investigated indicating that there is a trade-off between the speed of approaching the solution and stability of the swarm. In the future the algorithm will be extended for more complicated requirements and will be used in real-robot scenarios.

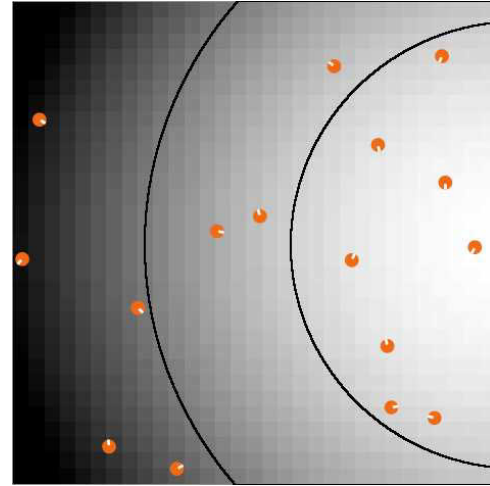


Figure 7: A screenshot of the robot arena. Three different task-regions are located based on their distances from the light source.

Time	0	20000	35000
$th_{1:2}$	25	25	25
$th_{2:3}$	75	75	50
Swarm Size	16	8	8

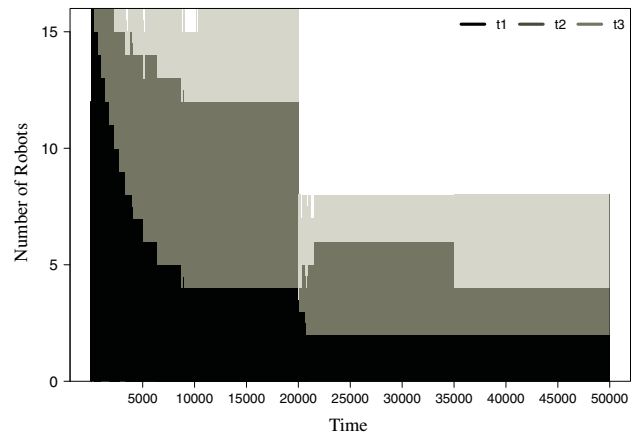


Figure 8: Medians of the number of robots in each task in 25 runs. The system starts with 16 robots while the demands for the three tasks are 25%, 50%, and 25% respectively. At time-step 20000, all the robots in the second task are removed from the arena. In time-step 35000, the demands change to 25%, 25%, and 50% respectively. Tasks are represented with  $t1$ ,  $t2$ ,  $t3$ .

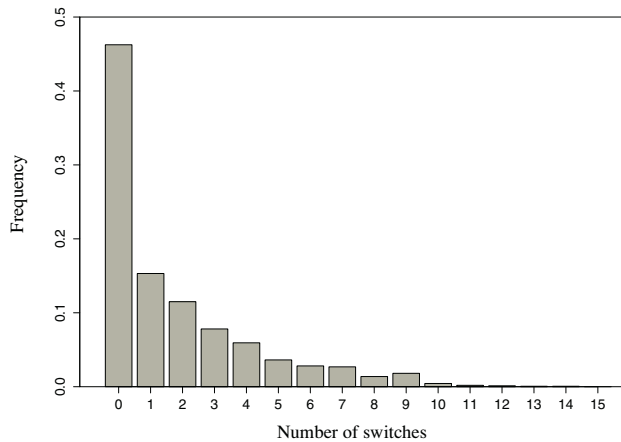


Figure 9: Frequency of number of switch-backs in the first 10000 time-steps for all the 16 agents and 25 runs.

### Acknowledgements

This work is supported by: EU-ICT project ‘CoCoRo’, no. 270382; EU-IST-FET project ‘SYMBRION’, no. 216342; EU-ICT project ‘REPLICATOR’, no. 216240; Austrian Federal Ministry of Science and Research (BM.W F); EU-ICT project ‘ASSISI.bf’, no. 601074.

### References

- Beshers, S. (2001). Social inhibition and the regulation of temporal polyethism in honey bees. *Journal of Theoretical Biology*, 213(3):461–479.
- Beshers, S. and Fewell, J. (2001). Models of division of labor in social insects. *Annu. Rev. Entomol.*, 46:413–440.
- Beshers, S., Robinson, G., and Mitternath, J. (1999). Response thresholds and division of labor in social insects. In Detrain, C., Deneubourg, J., and Pasteels, J., editors, *Information Processing in Social Insects*, pages 115–139. Birkhuser Basel.
- Bonabeau, E., Sobkowski, A., Theraulaz, G., and Deneubourg, J.-L. (1997). Adaptive task allocation inspired by a model of division of labor in social insects. In *Biocomputing and emergent computation: Proceedings of BCEC97*, pages 36–45. World Scientific Press.
- Gadagkar, R. (2001). Division of labour and organization of work in the primitively eusocial wasp *ropalidia marginata*. In *Proceedings of the Indian National Science Academy - Part B: Biological Sciences*, volume 6 of 67, pages 397–42.
- Gross, R., Nouyan, S., Bonani, M., Mondada, F., and Dorigo, M. (2008). Division of labour in self-organised groups. In *SAB*, pages 426–436.
- Hölldobler, B. and Wilson, E. (2008). *The Superorganism: The Beauty, Elegance, and Strangeness of Insect Societies*. W. W. Norton and Company, New York.
- Huang, Z.-Y. and Robinson, G. (1999). Social control of division of labor in honey bee colonies. In Detrain, C., Deneubourg, J., and Pasteels, J., editors, *Information Processing in Social Insects*, pages 165–186. Birkhuser Basel.
- Huang, Z. Y. and Robinson, G. E. (1992). Honeybee colony integration: worker-worker interactions mediate hormonally regulated plasticity in division of labor. *Proc Natl Acad Sci U S A*, 89(24):11726–9.
- Huang, Z. Y. and Robinson, G. E. (1996). Regulation of honey bee division of labor by colony age demography. *Behavioral Ecology and Sociobiology*, 39(3):147–158.
- Johnson, B. (2010). Division of labor in honeybees: form, function, and proximate mechanisms. *Behavioral Ecology and Sociobiology*, 64(3):305–316.
- Julian, G. E. and Cahan, S. (1999). Undertaking specialization in the desert leaf-cutter ant *acromyrmex versicolor*. *Anim Behav*, 58(2):437–442.
- Karsai, I. and Schmickl, T. (2011). Regulation of task partitioning by a “common stomach”: a model of nest construction in social wasps. *Behavioral Ecology*, 22:819–830.
- Labella, T. H., Dorigo, M., and Deneubourg, J.-L. (2006). Division of labor in a group of robots inspired by ants’ foraging behavior. *ACM Trans. Auton. Adapt. Syst.*, 1(1):4–25.
- Naug, D. and Gadagkar, R. (1999). Flexible division of labor mediated by social interactions in an insect colony a simulation model. *Journal of Theoretical Biology*, 197:123–133.
- Robinson, G. E. (1992). Regulation of division of labor in insect societies. *Annu Rev Entomol*, 37:637–65.
- Schmickl, T., Möslinger, C., and Crailsheim, K. (2007). Collective perception in a robot swarm. In Şahin, E., Spears, W. M., and Winfield, A. F. T., editors, *Swarm Robotics - Second SAB 2006 International Workshop*, volume 4433 of *LNCS*, Heidelberg/Berlin, Germany. Springer-Verlag.
- Seeley, T. D. (1982). Adaptive significance of the age polyethism schedule in honeybee colonies. *Behavioral Ecology and Sociobiology*, 11:287–293.
- Theraulaz, G., Bonabeau, E., and Deneubourg, J.-L. (1998). Response threshold reinforcement and division of labour in insect societies. In *In Proc. Royal Society of London*, volume 265 of 5, pages 327–332.
- Tofts, C. (1993). Algorithms for task allocation in ants. *Bull. Math. Biol.*, 55:891–918.
- White, T. and Helferty, J. (2005). Emergent Team Formation: Applying Division of Labour Principles to Robot Soccer. *Engineering Self-Organising Systems*, pages 180–194.
- Wilson, E. O. (1971). *The insect societies*. Belknap Press of Harvard University Press.
- Winston, M. (1987). *The biology of the honey bee*. Harvard University Press.
- Yang, Y., Zhou, C., and Tian, Y. (2009). Swarm robots task allocation based on response threshold model. In Gupta, G. S. and Mukhopadhyay, S. C., editors, *ICARA*, pages 171–176. IEEE.

# Cyclic Behavior in Gene-Culture Coevolution Mediated by Phenotypic Plasticity in Language

Tsubasa Azumagakito<sup>1,2</sup>, Reiji Suzuki<sup>1</sup> and Takaya Arita<sup>1</sup>

<sup>1</sup>Graduate School of Information Science, Nagoya University

<sup>2</sup>azumagakito@alife.cs.is.nagoya-u.ac.jp

## Abstract

The evolution of language has been the subject of much debate and speculation. It is difficult to study in a scientific manner and remains an open research question. This paper proposes an integrated computational framework for investigating possible scenarios of genetic and cultural evolution of language. Specifically, our framework aims to capture cultural evolution to allow for investigation phylogenetic dynamics of language, and at the same time to capture genetic evolution of phenotypic plasticity to allow for investigation of the role of the Baldwin effect in language evolution, while keeping the framework as simple as possible. In our evolutionary experiments and analysis, we discovered a coevolutionary scenario involving biological evolution of phenotypic plasticity, and a cyclic coevolutionary dynamic between genetic and cultural evolution, mediated by phenotypic plasticity.

## Introduction

Language distinguishes humans from other animals. Of course, other animals also engage in vocal communication. For example, Velvet monkeys can convey some simple information using alert calls (Sayfarth et al., 1980). However, such animals' vocal communication lacks the complex grammar and high expressiveness that characterizes human languages. Why do only humans have sophisticated language? This is one of the core questions on the path to understanding the human identity. This paper focuses on the evolution of the fundamental traits underlying communicative interaction in the context of biological evolution (genetic evolution of the language faculty), such as the rules or conventions for the effective communication necessary for collective behaviors. We assume that such traits can evolve under directional selection because the traits can be modified incrementally to increase the benefit from communicative interactions. We also assume such traits must be shared between individuals for communication to succeed. Accordingly, at least some of the selection will be positively frequency-dependent. This might obstruct evolution based on directional selection. We believe that this captures a fundamental and general problem in the evolution of communicative traits. For example, in the context of language

evolution it has been pointed out that mutations in grammar cannot be beneficial because the peers of an individual with a grammar mutation may not understand the mutant form (Pinker and Bloom, 1990; Glackin, 2010).

We believe that nature's solution to this challenging problem is found in the evolution of phenotypic plasticity. Phenotypic plasticity refers to the variability in the phenotype obtained from a given genotype resulting from development in different environments (West-Eberhard, 2003). In recent evolutionary biology, ontogenetic adaptation<sup>1</sup> based on phenotypic plasticity is recognized as one of the key factors that brings about adaptive evolution of novel traits (West-Eberhard, 2005; Gilbert and Epel, 2009). Wund provided a summary of eight hypotheses on how plasticity might influence evolution (including several pieces of empirical support), focusing mainly on adaptation to new environments (Wund, 2012). For example, the hypothesis that phenotypic plasticity promotes persistence in a new environment, and the hypothesis that a change in the environment can release cryptic genetic variation via phenotypic plasticity, in turn impacting the rate of evolutionary responses. Zollman and Smead (2010) analyzed simple models of language evolution based on Lewis's signaling game and the prisoner's dilemma game. They observed that the presence of plastic individuals alters the trajectory of evolution by directing the population away from a non-adaptive signaling and toward the optimal signaling. They termed this the "Baldwin optimizing effect". Suzuki and Arita also showed that such an adaptive shift can occur repeatedly, using a computational model of the coevolution of signal sending behavior and signal receiving behavior, that incorporated behavioral plasticity (Suzuki and Arita, 2008, 2012, 2013). These studies indicate that learning could be an important driving force for adaptive evolution in the context of communicative interactions.

This paper also looks at the relationships between two aspects of language evolution: biological evolution and cultural evolution. The relationship between genes and lan-

<sup>1</sup>Adaptive changes that occur during the lifetime of an organism (e.g., learning).

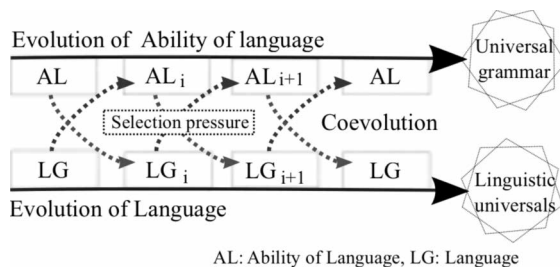


Figure 1: Coevolution between language and language ability. The upper arrow represents the evolution of language ability in general. The lower arrow represents the evolution of language itself. Language ability and language are represented by  $AL$  and  $LG$ , respectively.  $AL_i$  controls the selection pressures shaping  $LG_{i+1}$  and conversely  $LG_i$  controls the selection pressures shaping  $AL_{i+1}$ . Universal grammar (the theory proposing that the ability to learn linguistic grammar is hard-encoded into the brain) and Linguistic universals (general pattern that potentially exists in almost all of natural languages) may have emerged as an inevitable result of this coevolution.

guage is extremely complex and shrouded in controversy. Furthermore, rather than viewing language as a monolithic and independent entity, modern researchers typically break it down into its component mechanisms and analyze these independently (Fitch, 2011). Steels (2011) discussed various computational models of cultural evolution. He concluded that cultural evolution is a more powerful process than usually assumed, and that human language evolution's dependence on genetic evolution is relatively limited. Several researchers argue that cultural evolution has fundamentally limited influence on the genetic evolution of the language faculties. This contrasts with various results that indicate that genetic biases are essential to language evolution. For example, Chater et al. (2009) have shown using a computational model that there are strong restrictions on the conditions under which the Baldwin effect can embed arbitrary linguistic constraints, and that the effect only emerges when language provides a stable target for natural selection. These approaches should be seen as complementary. There is a need to integrate these efforts and explore the relevant gene-culture coevolutionary interactions (Mesoudi et al., 2011).

We suggest the insights from these studies of language evolution together can be brought together using the concept of coevolution between language and brain, as it lets us integrate biological and cultural evolution. The idea of coevolution was originally suggested by (Darwin, 1871), and others have taken up his lead (Deacon, 1997). As illustrated in Fig. 1, the main idea is as follows: On one hand, a language is continuously changed its users, which brings about the linguistic variation much like mutation brings about ge-

netic variation. Language variants that can easily be learned by their users can survive and thus spread in the population of languages. On the other hand, having innate linguistic abilities (e.g., universal grammar) that equip an individual to handle the existing language variants and language changes provide a fitness advantage. Thus, genes for an innate language faculty will spread in the biological population. So we have two intertwined adaptation processes: language adapts to the brain, and the brain adapts to language. We believe that this insight is crucial to a comprehensive understanding of language evolution.

In this paper, we employ the coevolutionary framework described above, and propose a bottom-up computational model for investigating possible scenarios of the genetic and cultural linguistic evolution. We aim to capture cultural evolution to investigate the phylogenetic dynamics of language evolution, while at the same time capturing genetic evolution of phenotypic plasticity. The latter allows us to investigate the role of the Baldwin effect (typically interpreted as a two-step evolution of the genetic acquisition of a learned trait without the Lamarckian mechanism (Peter Turney and Anderson, 1996)) in language evolution, all the while keeping the framework as simple as possible. In order to do this, we extend our previous works on language evolution (Suzuki and Arita, 2008, 2012; Azumagakito et al., 2011, 2012). The main idea is to express the linguistic space as a polar coordinate system, in which individuals and languages gradually move about by genetic evolution and cultural evolution, respectively, as follows: 1) the more an individual can communicate, and the higher the expressivity of its languages, the more offspring it can produce. This nudges the population toward the locations of fit individuals. 2) Languages move towards the center of their user base, and divide, merge or go extinct in accordance with the distribution of their users.

Recently, there is much research on the diversification processes of real human language (Pagel, 2009; Lupyán and Dale, 2010; Atkinson et al., 2008; Atkinson, 2011). Levinson and Gray (2012) reviewed various methods to investigate the diversification of languages, and observed that phylogenetic analysis techniques from evolutionary biology are useful tools for elucidating the diversification of language. Phylogenetic patterns of language evolution were uncovered using such analyses, but the causal mechanisms of diversification are still unclear. These previous investigations proceed within the observational and deductive realm. The framework proposed here provides the means for experimentation, and a method to generate phylogenetic trees that can help elucidate the causal mechanisms of language diversification (see Fig. 2).

2) Analyze the feature of estimated tree, for example word-order relation between language families.



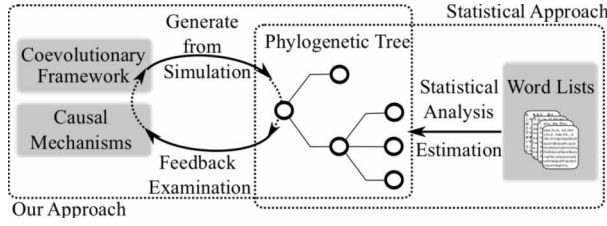


Figure 2: Phylogenetic comparative analysis using the proposed framework. Simulation results based allow us to generate phylogenetic tree of language families. By comparing the experimentally generated trees with the trees inferred from real linguistic data sets (such as standard word-lists), we can elucidate how and what causal mechanisms may have brought about the features of the inferred trees, and verify the validity of our simulation model. In this paper, we generate a phylogenetic tree from an evolution experiment. A comparison of it with the inferred trees from the real linguistic data is our future work.

### Model

We propose an integrated computational framework for investigating possible scenarios of genetic and cultural evolution of language. This framework allows us not only to capture coevolutionary interactions between languages and agents, but also to track the phylogenetic evolutionary process of languages.

### The Linguistic Space

There are  $N$  agents in a population, and they can communicate with each other using their shared languages. Agents and languages exist in a two-dimensional linguistic space, represented as a polar coordinate system, as shown in Fig. 3. Each language ( $L$ ) is defined as a point in the space. The distance  $r_L$  from a language to the coordinate system's origin represents the language's expressiveness, which contributes to the expected fitness benefit of a successful communication in that language. The angle of the language  $\theta_L$  to the origin represents its structural character. Each agent ( $A$ ) is represented as a point and a field surrounding the point. This point represents the agent's innate language ability, determined by its genotypes  $r_A$  and  $\theta_A$ . The agent can use the corresponding language in the linguistic space without learning. The field represents its linguistic plasticity (i.e. the range of its linguistic learning ability), as a  $p_A \times p_A$  fan-shaped field determined by its genotype  $p_A$ . The agent can use any language that falls within its plasticity field for communication. This polar coordinate system captures the fact that, as expressivity increases, the space of possible linguistic structures grows, such that agents with more expressive languages will be harder to communicate successfully with, due to the limited size of the plasticity field.

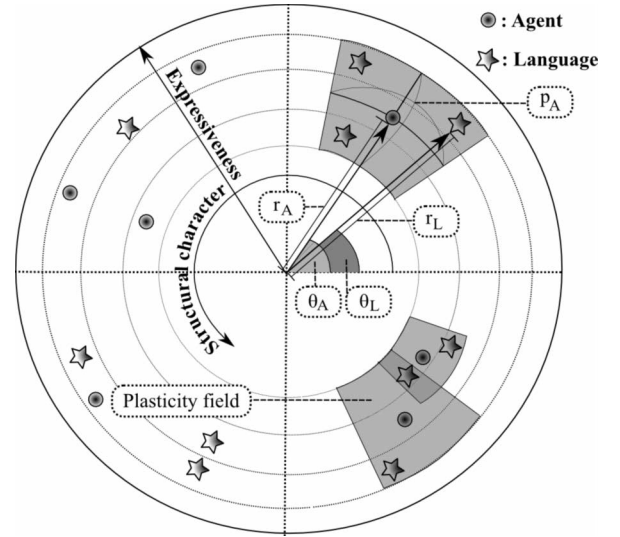


Figure 3: The linguistic space.

### Linguistic Interactions

In each generation, all possible pairs of agents make an attempt to communicate. If the two agents of a pair share one or more languages, they can communicate successfully. The fitness of each agent depends on its number of successful communicative interactions, the expressiveness of the languages used in those interactions, and the cost of its linguistic plasticity. The fitness function is defined as:

$$Fitness = N_{ca}^{w_1} \cdot \left( \frac{\sum_{i=0}^{N_{ul}-1} r_{L_i}}{N_{ul}} \right)^{w_2} - (p_A^2)^{w_3}, \quad (1)$$

where  $w_i$  ( $i=1, 2$  and  $3$ ) are weights for the three components of the fitness function. The first component represents the benefit from successful communicative interactions, which is proportional to the number of agents with which the focal agent successfully communicated  $N_{ca}$ . The second component represents the benefit from the expressivity of the languages available to the agent.  $N_{ul}$  is the number of languages within the plasticity field of the focal agent, and  $r_{L_i}$  is the expressiveness of the  $i$ -th language among them. The second component is an approximation of the average expressiveness of the languages used for successful communications. We adopted an approximation in order to reduce the computational cost. The third component represents the cost of linguistic plasticity. It is proportional to the area of plasticity ( $p_A^2$ ), because the further a language is removed from an agent's innate language ability, the bigger the demands on the agent's plasticity, and thus plasticity for further and further removed languages comes with higher and higher costs for maintaining the requisite plasticity. Overall, this fitness definition reflects that agents who can communicate with many other using expressive languages acquired

with limited linguistic plasticity will have high fitness.

### Biological Evolution of Language Ability

After the communicative interactions of agents, biological evolution of language ability occurs as follows: 1) Parent agents for the next generation are selected using roulette wheel selection (i.e. the probability that an agent is picked as a parent is proportional to its fitness). 2) Each genotype of each offspring is mutated with probability  $P_m$ . Mutation adds a small random value of  $R(0.01)$  to the values of  $r_A$  and  $p_A$ , and of  $R(0.01/r_A)$  to  $\theta_A$ , where  $R(x)$  produces random numbers between  $-x$  and  $x$  with a triangular distribution. Note that the range of a random value for  $\theta_A$  is inversely proportional to  $r_A$  of the parent. This keeps displacement of the innate language ability of an agent and the change in plasticity independent from the location of the agent in linguistic space.

### Cultural Evolution of Language

Subsequently, the language population evolves according to four cultural processes: extinction, cultural change, division and fusion.

**Extinction** Any language that was not used by any agents in this generation do not appear in the next generation, and thus are removed from the linguistic space. This models the extinction of unused languages.

**Cultural change** The users of a language change its characteristics. We model a cultural change of language as a change in the location of the language within the linguistic space. Each agent creates, for every language it used, an attraction force (vector)  $F$  that drags the languages toward the agent's location, as shown in Fig. 4(a). The length  $F$  is  $f_{max}/N_{ul}$ , where  $f_{max}$  is the parameter that determines the maximum length of  $F$  and  $N_{ul}$  is the number of languages within the plasticity field of the agent. Each language moves to the location determined by the resultant vector of all the forces exerted on it. Fig. 4(a) shows an example of cultural change of language.

**Division** A language is divided into two languages if the forces of cultural change strongly pull it in opposing directions. Fig. 4(b) shows an example process of a language division. To determine the direction of a division of a language, we adopt "Principal Component Analysis" (PCA) (Pearson, 1901). Because the first component axis of PCA on the forces working on a language corresponds to the direction of the forces' maximum variance, we use the second component axis ( $L$ ) to divide the forces into two groups. We calculate the resultant force for each group. A division process occurs when the length of either or both resultant forces is larger than the threshold parameter  $Div_f$ . These resultant forces are then used to determine the locations of the two languages resulting from the division.

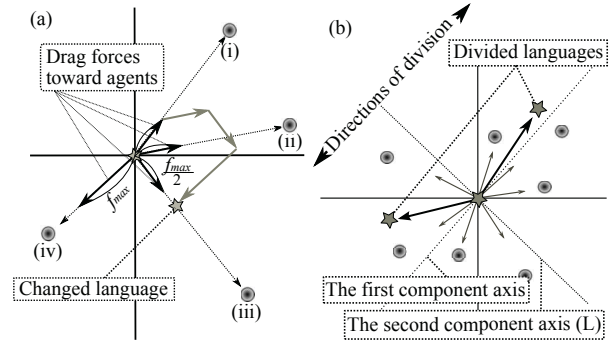


Figure 4: Panel (a) shows an example of a cultural evolution step of a language. Note that agents (i), (ii) and (iii) can each use another language (not shown) besides the focal one. Therefore, their pulling forces are only half that of agent (iv). Panel (b) shows an example of a language division process.

**Fusion** When the distance between two languages is close enough, these languages are united into one language. This process occurs when two languages' difference in distance to origin and angle to origin are smaller than the thresholds  $\beta$  and  $\gamma$  respectively.

Through the above processes, the agent population and the language population coevolve.

### Simulation Results

We conducted evolutionary experiments for 30000 generations and visualized the results in the linguistic space. The following parameters were used:  $N=2000$ ,  $w_1=1$ ,  $w_2=1$ ,  $w_3=30$ ,  $P_m=0.01$ ,  $Div_f=0.00003$ ,  $\beta, \gamma=0.001$  and  $f_{max}=0.00003$ . The initial values of  $r_A$  and  $r_L$  were picked at random from  $[0, 0.001]$ .  $\theta_A$  and  $\theta_L$  were picked at random from  $[0, 2\pi]$ .

Fig. 5 shows an example run of this experiment. We observed a typical evolution scenario, which we summarize in Fig. 6. From the initial population, both agents and languages are aggregated around the origin of the linguistic space. At this stage agents communicated successfully with each other using just a few languages at around the origin, because their innate linguistic abilities were quite similar. From there onward, we observed that (0) the number of languages rapidly increased until the 250th generation, reached to around 40 languages. This could be interpreted as a "Linguistic burst". This is thought to be due to the high concentration of agents around the origin easily leads to opposing cultural pull and hence frequent language division.

After the 250th generation, we observed cyclic coevolution processes of languages and agents. Let us look at the evolution process from the 7500th to the 13500th generation (i-iii) as example. Around 7500th generation, we see agents with smaller plasticity fields clustered densely together. In

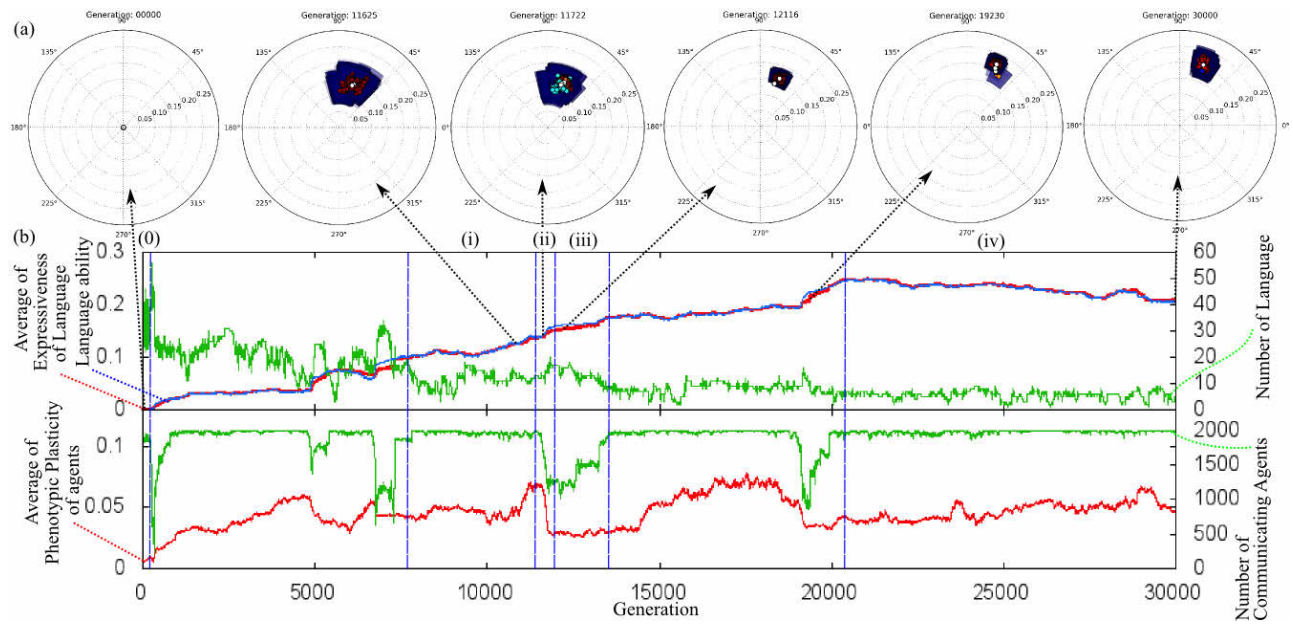


Figure 5: An example evolution process. (a) Several snapshots of coordinate distributions of languages and agents. Each agent is shown as a colored circle, with its color representing its fitness, and a fan-shaped field representing its linguistic plasticity. Languages are shown as white circles. (b) Evolution plots. The x-axis shows the generation number.

this situation, there is only weak selection pressure on innate language ability, because agents can already communicate successfully. This lack of selection leads innate language abilities to scattered by neutral evolution. This in turn leads to a gradual increase in phenotypic plasticity until around the 11500th generation (i), because additional plasticity became necessary for the agents to keep their communication successful. Then, around the 11500th generation, some agents with more expressive innate language ability and lower phenotypic plasticity appeared (ii), and occupied the population quickly. Instead of communicating with many agents in less expressive languages while incurring high plasticity costs, these agents communicate with a limited number of neighbors using more expressive languages while incurring only a small plasticity cost, which results in a net relative fitness gain. Thus, the average expressiveness of the innate language ability became larger than that of the existing languages, and the number of successful communications decreased drastically. Note that the number of languages increased because languages were dragged by two groups: the group of agents with more expressive ability of language and the group of the agents with less expressive ability of language.

After that, from the 11800th generation until the 14700th generation, the language population evolved toward the languages used by these adaptive agents, via a process of cultural evolution arising from the increased use of the more

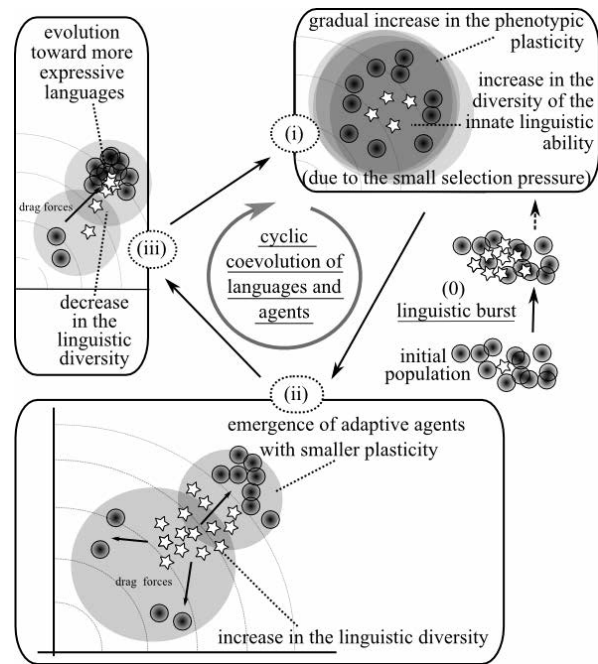


Figure 6: Typical scenario of evolution process.

expressive languages (iii), increasing the number of successful communications among agents again. Languages now



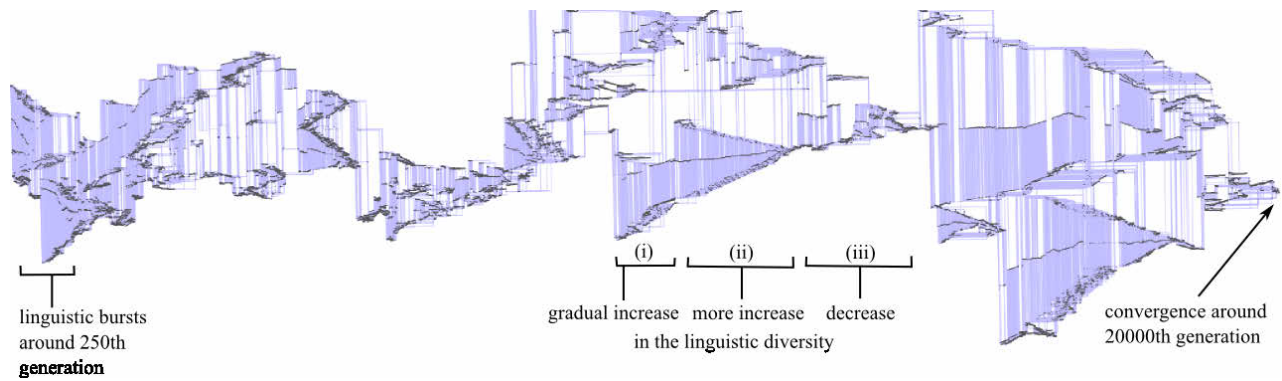


Figure 7: The generated phylogenetic network of languages (from the 250th to the 20000th generations). Each black node represents a language, and each link between the left node and the right node represents the genealogical connection between the ancestral language and its offspring language. There are 64389 nodes and 91976 links in this tree. The language division occurred 45838 times and fusion of language occurred 23802 times. The death of language occurred 22036 times.

distant from the agents' (shifted) innate language abilities go extinct, leading to a gradual decrease in the number of languages. As a result, language expressivity caught up with the expressivity of innate language ability, i.e. both the agent population and language population moved outward from the origin, and the overall system state was back at the initial state of a cycle. As this cyclic coevolution processes repeated, the expressiveness of languages and agents' innate language ability increased, while overall the number of languages decreased. This could be interpreted as the emergence of major dominant languages (Abrams and Strogatz, 2003). The cyclic mechanism can be summarized as follows: (i) The phenotypic plasticity of the agents increases gradually due to the selection for the robustness of successful communications against the increased genetic variation of the innate language ability by a genetic drift (caused by the previous step). (ii) Some agents with the more expressive innate language ability and the smaller plasticity occupy the population quickly because they can communicate using more expressive languages than other agents with the larger plasticity. (iii) The expressiveness of languages increases and the diversity of languages decreases because the languages are dragged by the agents with more expressive language ability. This brings back to the process to the beginning, because the small fitness differences due to the very few number of different languages creates the variation of the innate language ability of agents.

However, (iv) the evolution process eventually halts once the expressiveness of languages reached the high value of 0.2 after the 20000th generation. This is thought to be due to the fact that it became increasingly difficult for agents with relatively high expressiveness to maintain enough plasticity for successful communications, as the increasing cost cancels out the benefit of their expressivity.

In addition to the above analysis, we show the phyloge-

netic network of languages for our basic simulation experiment (Fig. 7). We found that a process of diversification and unification of languages emerged through repetition of the interaction processes between genetic and cultural evolution that we described in the previous section. In previous studies of language change, the phylogenetic relations of language families is generally represented as a tree (Gray and Atkinson, 2003). Fusion of languages cannot be captured using a tree representation. However the generated network showed that language fusion occurred frequently. This shows the important role the cultural processes of fusion plays in the evolution of language within our model.

Finally, we conducted experiments to study the effects of the model's parameters on the evolution process. First, to investigate the effect of learning cost, we conducted experiments with various settings of the weight on the learning cost,  $w_3$ . We found that the duration until the population reached the coevolution phase increased as  $w_3$  increased. A higher cost of learning puts the population under stronger selection pressure for low plasticity. Because individuals with the low plasticity were less robust against mutations and often failed to leave offspring, evolution speed dropped. Also, the speed of the increase in the expressiveness of languages was inversely proportional to  $w_3$ , due to the increased duration until the start of the coevolutionary phase. For example, in the case of no cost ( $w_3 = 0$ ), the duration was quite short: the coevolutionary phase started after about one hundred generations. In the case of a huge learning cost ( $w_3 = 100000$ ), the evolution of language and population stagnated around the origin, because individuals could not increase their plasticity at all. It also should be noted that higher values of  $w_3$  lead to shorter cycle period. This is thought to be due to the fact that the rapid decrease in phenotypic plasticity (ii) tends to occur more often as the cost



of plasticity increases.

We also investigated the effects of  $f_{max}$  which determines the strength of the attraction force that pulls a language toward its users. Because this parameter is used in the processes of cultural change and division of languages, we assumed the condition in which the threshold for the division  $Div_f$  was proportional to  $f_{max}$  ( $Div_f = f_{max} * 300$ ) in order to mainly focus on the effects of change in  $f_{max}$  on the process of cultural change. Experiments with different settings of  $f_{max}$  (from  $1.0 * 10^{-7}$  to  $1.0 * 10^{-4}$ ) showed that the chances of all languages dying off during the early generations increase with increasing  $f_{max}$ . This is because at large  $f_{max}$ , when there are many individuals pulling on a language, the resultant of the attraction forces tend to be so large that it displaces the language outside the plasticity ranges of the agent population. However, once the initial increase in the number of languages (0) has started successfully, we see a more rapid increase in the expressiveness of languages in the cyclic processes, because the larger pulling force facilitates rapid adaptation of in the language population. Especially in trials at high  $f_{max}$  ( $1.0 * 10^{-5}$ ), successful evolution was only observed when by chance the initial population had high plasticity. At extremely high  $f_{max}$  ( $1.0 * 10^{-4}$ ), all trials failed in about 10 generations.

### Discussion

Cultural linguistic change is often assumed to be much faster than biological change. Chater et al. (2009) showed, using a computational model, that genetic natural selection may not keep pace with a language change. Conversely, Számadó and Szathmáry (2012) argued that there are many ways for organisms to adapt to quickly changing targets, and showed numerous examples of rapid evolutionary change. For example, the rate of biological adaptation depends on the population size and genetic variation. This means that there is a possibility that biological adaptation too can be quick. Also, the phenotypic plasticity of a genotype effectively could facilitate adaptation, via the Baldwin effect. On the other hand, they also pointed out the possibility that the rate of language change could slow down. Historically the rate could have been much slower than it is now, due to smaller population sizes, slower rates of technological innovations, more limited contexts of language use, and much smaller vocabulary. Furthermore, the rate of linguistic change depends on frequency of use: words and rules used more frequently evolve far slower.

The experimental results obtained using our framework of genetic and cultural evolution of language demonstrates that the rate of language evolution could change through cyclic coevolutionary processes. This partly support Számadó et al.'s claims, especially with regards to the way phenotypic plasticity promotes adaptation. In addition to Számadó et al.'s claims, we obtained the following insights from our simulation: 1) Diversity across language groups increases

fitness variance, which accelerates the rate of biological evolution. 2) The rate of cultural evolution tends to be restricted by the plasticity of individuals, as languages cannot survive outside of the linguistic plasticity range of individuals. 3) The rate of cultural change can be slow, especially when individuals reduce their learning cost as they cluster around existing languages with sufficient expressiveness for communication. In contrast to situations with no linguistic conventions among speakers, this tends to lead language evolution to stagnate. We think that the rate of cultural change may be faster when there are no linguistic conventions among speakers, and slower when some shared conventions exist among them.

### Conclusion

This paper proposed an integrated framework for investigating genetic and cultural evolution of language. On basis of this framework, we constructed an agent-based model capturing both cultural evolution of languages and biological evolution of linguistic faculties, expressed on a two-dimensional linguistic space. Our evolutionary experiments showed that, after an initial rapid increase in the number of languages, a cyclic coevolution process occurs in which biological evolution and cultural evolution proceed in alternation. Here we observed genetic assimilation of language into innate linguistic ability. Eventually, the population reached languages with high expressiveness. Our model can be regarded as an "emergent computational thought experiments" (Bedau, 1999), or "opaque thought experiments" in which the consequences follow from the premises in such a non-obvious manner that the consequences can only understood through systematic enquiry (Di Paolo et al., 2000). We believe our model can also be extended to function as a realistic simulacra type of simulation model. For this purpose, we should further investigate what parameter settings correspond best to reality, especially in regard to the relative speeds of cultural and biological evolution.

### References

- Abrams, D. M. and Strogatz, S. H. (2003). Modelling the dynamics of language death. *Nature*, 424:900.
- Atkinson, Q. D. (2011). Phonemic diversity supports a serial founder effect model of language expansion from Africa. *Science*, 332(6027):346–349.
- Atkinson, Q. D., Meade, A., Venditti, C., J. Greenhill, S., and Pagel, M. (2008). Languages evolve in punctuational bursts. *Science*, 319(5863):588.
- Azumagakito, T., Suzuki, R., and Arita, T. (2011). Visualizing language evolution as an emergent phenomenon based on biological evolution and learning. *Artificial Life and Robotics*, 16(3):366–372.
- Azumagakito, T., Suzuki, R., and Arita, T. (2012). A simple integrated framework for investigating genetic and cultural evolution of language. In Scott-Phillips, T. C., Tamariz, M., Cartmill, E., and Hurford, J., editors, *The evolution of language* :

- proceedings of the 9th international conference on the evolution of language (EVOLANG 9)*, pages 405–406. World Scientific Publishing, Singapore.
- Bedau, M. A. (1999). Can unrealistic computer models illuminate theoretical biology? In *Proceedings of the 1999 Genetic and Evolutionary Computation Conference Workshop Program*, pages 20–23.
- Chater, N., Real, F., and Christiansen, M. H. (2009). Restriction on biological adaptation in language evolution. *Proceedings of the National Academy of Sciences*, 106:1015–1020.
- Darwin, C. (1871). *The Descent of Man, and Selection in Relation to Sex*. John Murray.
- Deacon, T. W. (1997). *The symbolic species: The co-evolution of language and brain*. W. W. Norton & Company, Inc.
- Di Paolo, E. A., Noble, J., and Bullock, S. (2000). Simulation models as opaque thought experiments. In *Proceedings of the Seventh International Conference on Artificial Life (ALife XI)*, pages 497–506.
- Fitch, W. T. (2011). Biological versus cultural evolution beyond a false dichotomy comment on “modeling the cultural evolution of language” by Luc Steels. *Physics of Life Reviews*, 8(4):357–358.
- Gilbert, S. F. and Epel, D. (2009). *Ecological Developmental Biology: Integrating Epigenetics, Medicine, and Evolution*. Sinauer Associates, Sunderland, MA.
- Glackin, S. N. (2010). Universal grammar and the Baldwin effect: a hypothesis and some philosophical consequences. *Biology and Philosophy*, 26(2):201–222.
- Gray, R. D. and Atkinson, Q. D. (2003). Language-tree divergence times support the anatolian theory of Indo-European origin. *Nature*, 426:435–439.
- Levinson, S. C. and Gray, R. D. (2012). Tools from evolutionary biology shed new light on the diversification of languages. *Trends in Cognitive Sciences*, 16(3):167–173.
- Lupyan, G. and Dale, R. (2010). Language structure is partly determined by social structure. *PLoS ONE*, 5(1):e8559.
- Mesoudi, A., Mcelligott, A. G., and Adger, D. (2011). Introduction: Integrating genetic and cultural evolutionary approaches to language. *Human Biology*, 83(2):141–151.
- Pagel, M. (2009). Human language as a culturally transmitted replicator. *Nature Reviews Genetics*, 10(6):405–415.
- Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine A*, 2(6):559–572.
- Peter Turney, D. W. and Anderson, R. (1996). Evolution, learning, and instinct: 100 years of the Baldwin effect. *Evolutionary Computation*, 4(3):4–8.
- Pinker, S. and Bloom, P. (1990). Natural language and natural selection. *Behavioral and Brain Sciences*, 13(4):707–784.
- Sayfarth, R. M., Cheney, D. L., and Marler, P. (1980). Monkey responses to three different alarm calls: Evidence of predator classification and semantic communication. *Science*, 210:801–803.
- Steels, L. (2011). Modeling the cultural evolution of language. *Physics of Life Reviews*, 8(4):339–356.
- Suzuki, R. and Arita, T. (2008). How learning can guide evolution of communication. In *Proceedings of the Eleventh International Conference on Artificial Life (ALife XI)*, pages 608–615. MIT Press.
- Suzuki, R. and Arita, T. (2012). Reconsidering language evolution from coevolution of learning and niche construction using a concept of dynamic fitness landscape. In McCrohon, L., Fujimura, T., Okanoya, K., Fujita, K., Suzuki, R., Martin, R., and Yusa, N., editors, *Five Approaches to Language Evolution (Proceedings of the Workshops of the 9th International Conference on the Evolution of Language (Evolang IX))*, pages 104–113. Evolang9 Organizing Committee.
- Suzuki, R. and Arita, T. (2013). A simple computational model of the evolution of a communicative trait and its phenotypic plasticity. *Journal of Theoretical Biology*, 330(7):37–44.
- Számádó, S. and Szathmáry, E. (2012). Evolutionary biological foundations of the origin of language: the co-evolution of language and brain. In Tallerman, M. and Gibson, K. R., editors, *The Oxford Handbook of Language Evolution*. Oxford university press, Oxford.
- West-Eberhard, M. J. (2003). *Developmental plasticity and evolution*. Oxford University Press.
- West-Eberhard, M. J. (2005). Phenotypic accommodation: adaptive innovation due to developmental plasticity. *Journal of Experimental Biology*, 304(6):610–618.
- Wund, M. A. (2012). Assessing the impacts of phenotypic plasticity on evolution. *Integrative and Comparative Biology*, 52(1):5–15.
- Zollman, K. J. S. and Smead, R. (2010). Plasticity and language: an example of the Baldwin effect? *Philosophical Studies*, 147(1):7–21.

## Detecting regime shifts in artificial ecosystems

Vasthi Alonso Chávez<sup>1</sup>, C. Patrick Doncaster<sup>1</sup>, John A. Dearing<sup>2</sup>, Rong Wang<sup>2</sup>,  
Jing-Lun Huang<sup>1</sup> and James G. Dyke<sup>3</sup>

<sup>1</sup>Centre for Biological Sciences, University of Southampton, Southampton SO17 1BJ, UK

<sup>2</sup>Palaeoecological Laboratory, Geography and Environment, University of Southampton, Southampton SO17 1BJ, UK

<sup>3</sup>Institute for Complex Systems Simulation, University of Southampton, Southampton, SO17 1BJ, UK

### Abstract

Ecosystems are subjected to a range of perturbations that have the potential to induce relatively sharp transitions in states. These can be referred to as regime shifts or critical transitions. They may be driven by perturbations that vary over a wide range of spatial and temporal scales, from responses to deforestation within a small field to responses to the gradual increase of carbon dioxide in the Earth's atmosphere. Here we investigate potential early warning signals that may presage regime shifts in model ecosystems. We hypothesise and model a relationship between biodiversity and community structure that influences ecosystem structure. We argue that Artificial Life methodologies have potential to make substantial contributions to efforts searching to predict large changes in ecosystems and other elements in the Earth system, as there is a recognised limitation in empirical data and ability to conduct experiments in the real-world. Consequently simulation and exploration of the low-level mechanisms that give rise to regime shifts in artificial in-silico ecosystems represents a useful line of enquiry.

### Introduction

The relationship between complexity and stability is a well established topic in ecology (McCann 2000). Much of this discourse stems from May's seminal paper that found an inverse relationship between diversity and stability in simple model ecosystems (May 1973). As the number of linear connections between species increased, the probability that the ecosystem would be stable decreased.

Here we explore to what extent the complexity of artificial systems, and by extension real-world ecosystems, can be used to provide early warning signals of impending large changes in their states - sometimes referred to as regime shifts. That is, we do not explore the (causal) relationships between complexity and stability, rather we seek to leverage the maximum amount of information from very often complex systems in order to give an indication of how stable the system is to perturbations and how close they are to collapse.

Dynamical systems theory has been applied to a range of real-world systems in order to determine to what extent they exhibit bistability and the potential to undertake catastrophic fold bifurcations, and thus rapid changes from one attractor

to another (Scheffer et al. 2001). One example of this phenomenon is the eutrophication of a lake. When a lake is subject to nutrient loading, changes may occur from a state of clear water with rich submerged vegetation to a turbid one without submerged vegetation and loss of animal biodiversity. At early stages, water clarity seems little affected by increased nutrient concentrations until a critical threshold is reached. Once this threshold is passed, the lake abruptly shifts to a turbid stable state.

Given the range of perturbations facing ecosystems and an increasing awareness of their contribution to human well being in the terms of the services they provide, a great deal of research effort is being devoted to formulate robust early warning signals (Scheffer et al. 2012; Lenton et al. 2012). Leading indicators of impending regime shifts have been identified that operate regardless of the underlying processes driving the change (Scheffer et al. 2009; Carpenter and Brock 2006; Guttal and Jayaprakash 2008). Some studies however, have identified the potential for false positives or even an absence of signals (Hastings and Wysham 2010), and studies of the drivers of regime shift reveal other signals in community composition (Scheffer and Nes 2007). The sooner a warning can be detected, the more time is available to act on it. The challenge is to find signals that warn at the earliest opportunity but do not provide false alarms.

Our ability to conduct experiments on real-world ecosystems is often very limited. For example, studies of regime shifts in the Lake Erhai, Yunnan Province, in China involved gathering then analysing sediment cores along with the integration of large amounts of socio-economic data (Wang et al. 2012). At this scale, it is effectively impossible to manipulate ecosystem variables let alone ensure control systems are available to compare experimental results. One way to increase our understanding in this area is to perform simulation experiments on artificial ecosystems. Such simulations can be understood as conceptual models that produce data appropriate for statistical analysis and so allow the development of proof of concepts and initial hypothesis testing.

Our starting assumption is that regime shifts in real-world ecosystems are often a consequence of changes in the abun-

dance of keystone species. A keystone species is a species that has large effects on its environment and consequently plays a key role in the maintenance of the structure and integrity of its ecosystem. The loss of a keystone species will be expected to produce changes in a large number of other species such that the entire ecosystem may transition to a new state: a regime shift.

If we seek to undertake interventions and so avoid future regime shifts, waiting to observe changes in keystone species may be too late as the transition is already in progress. We propose instead monitoring properties of ecosystems that includes measures of complexity and structure. The novelty of our method is in using as detectors the species most vulnerable to extinction, which we term the ‘ecosystem canaries’, whose response to the perturbation precedes the approach to a critical tipping point. Provided they are correctly identified as sensitive to the driver of environmental change, they are not prone to false alarms. Provided they are not keystone species, their signal precedes those of the leading indicators for imminent critical transition.

### Nestedness and biodiversity analyses

We hypothesise that biodiversity has a relationship with ecosystem structure that depends on important properties of the dynamical system. In this section we define ways to measure both the biodiversity and the structure of a community. Then we correlate these measurements over time to investigate the relationship between these quantities under different stress regimes.

#### Hill’s biodiversity analyses

Species diversity has two main components: abundance of each species and species richness (Magurran and Magurran 1988). Hill’s biodiversity indices incorporate these two components in a single value and are easy to interpret ecologically. The index value increases when either the number of species or the evenness of the community increases. In this study, we use Hill’s biodiversity index  $N_2$ ,

$$N_2 = \frac{1}{\sum_{i=1}^S p_i^2}, \quad (1)$$

which is the inverse of the commonly used Simpson’s diversity index  $\sum_{i=1}^S p_i^2$  (Simpson 1949). Here  $S$  is the number of species and  $p_i$  is the proportion of the  $i$ th species in the community (Hill 1973).

Simpson’s diversity index indicates the probability of finding an already observed individual by randomly choosing it from the community (Simpson 1949). Hence, the larger the value of Simpson index, the less diverse the community will be. Contrastingly, the higher the value of Hill’s biodiversity index  $N_2$ , the more diverse the community will be.

### Nestedness analyses

If we assume that a community is an assemblage of meta-communities, where each meta-community contains a subset of the species found in richer meta-communities, then we can ask: how much information about the structure of the community can we obtain when we observe a single species? The answer is potentially a surprising amount, if we also understand how nested the community is.

The concept of nestedness was first introduced as an explanation of insular faunal structure where species abundances decrease with distance of islands from a continent and species in distant islands are a subset of species in proximate ones (Patterson and Atmar 1986). Latterly, the concept has been applied extensively to terrestrial communities (Cutler 1991; Fischer and Lindenmayer 2005).

Nestedness is a characteristic of the interconnectedness of species in a community (Ulrich et al. 2008). If a community is completely nested, the community structure and composition may be predicted entirely from the presence of the least abundant species; if it is completely un-nested, its composition is unpredictable, as in the case of communities with a high flux of species.

In general, nestedness studies focus on the analysis of spatial data (Rodríguez-Gironés and Santamaría 2006), or spatial analyses replicated through time (Timi and Poulin 2008; Heino et al. 2009). However, nested temporal assemblages can occur when most species respond similarly to inter-annual variation in conditions. In contrast, assemblages might be non-nested when different sets of species occur in different years (Elmendorf and Harrison 2009). Here we develop a novel approach in which measures of nestedness are computed for communities that change only temporally and not spatially (see Fig. 1).

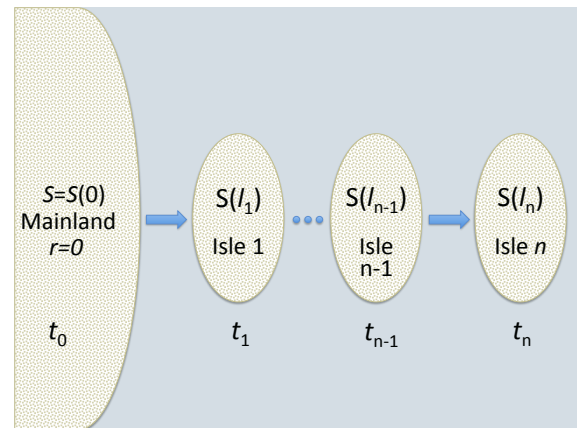


Figure 1: Nestedness in the structure of insular communities shows decreasing species abundance  $S$  with distance from mainland. Island  $S(I_n)$  contains a subset of the species found in previous islands so  $S(I_n) \in S(I_{n-1}) \in \dots \in S$ . Analogously, a given community  $S(t_n)$  at  $t = t_n$  can be seen as a subset of the community  $S(t_0)$  at  $t = 0$ , such that  $S(t_n) \in S(t_{n-1}) \in \dots \in S(t_0)$ .



One way to measure the nestedness of a community is via its incidence temperature (Patterson and Atmar 1986). This temperature ranges from 0 (completely nested) to 100 (completely un-nested), and provides a measure of the species richness across non-chronological times with respect to their incidence. The use of the term “temperature” can be understood from physical analogy: as the temperature of a collection of molecules increases then this additional energy will typically lead the molecules to occupy a larger volume. Knowing the position of a single molecule in a low temperature systems provides more information about the location of all the other molecules than higher temperature systems.

Originally, the nestedness temperature was presented as a measure of the degree of uncertainty in species extinction order and was linked in perfect analogy with the entropy of a system (Atmar and Patterson 1993). Recently, it has been argued that the nestedness temperature does not increase with extinction disorder (Almeida-Neto and Ulrich 2011) and therefore it cannot be explained as the inverse of the entropy (Almeida-Neto et al. 2007). However, nestedness temperature does give a measure of the community structure dependent on the community distribution. In this sense, nestedness temperature may be measured in non-dimensional units of entropy where a completely structured community would have the lowest entropy while completely disordered communities would have maximum entropy values.

## Hypothesis

According to Montoya et al. (2006) the interaction between species in an ecosystem may be so complex as to be impossible to understand. Here we argue that, although interactions between species in a community are very complex, the underlying mechanism of their dynamics are reflected in the structure and biodiversity of the system. If we think of an ecosystem as a complex network where each species may (or may not) be associated to other species, the structure of this network will certainly affect the function of the system (Strogatz 2001). Inter-species connections may be trophic in the sense of who eats whom, or competitive in the sense of who competes for whom over a food or resource. They may also emerge from ecosystems engineering effects (Jones et al. 1994) where one species alters the environment for another. As we are motivated to produce a general hypothesis in which if the structure of a community at a given time is affected by its biodiversity, we deliberately do not prescribe on the type of interactions among species. Instead, we propose that a correlation between nestedness temperature and Hills Biodiversity index may exist.

We assume that a community is composed of different types of species among which we can find keystone species, interacting species and “canary species”. The keystone species are those with a large effect on the system, the interacting species will also be linked to other species in the

community, while the presence or absence of canary species will have little impact. Canary species are also the most vulnerable to extinction. Consequently they will be rare in the community and will have little or no interaction with other species (see Fig. 2).

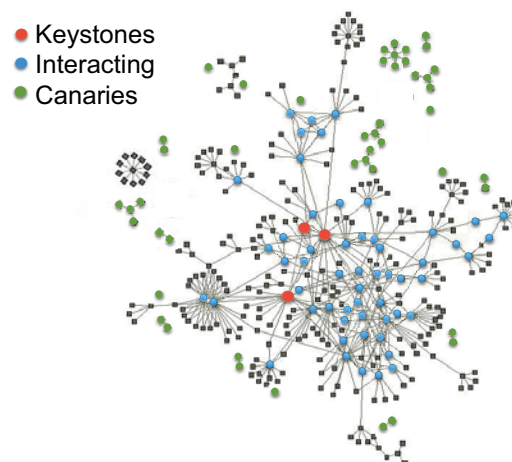


Figure 2: A community of species represented by their interactions with each other. Red dots represent keystone species, which are the species with the most links in the community. Blue and black dots are interacting species with fewer links to other species. Green dots are canary species, which are relatively isolated from the community and have very few links with other species. The lines between dots correspond to the links between species. Modified from Krebs (2012).

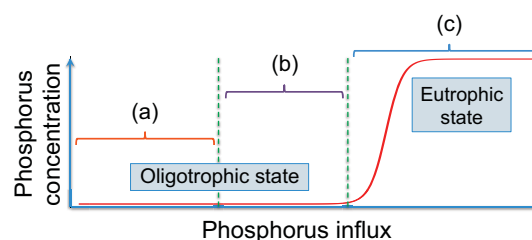


Figure 3: Increase of phosphorus influx into a lake drives the system from an oligotrophic state (clear water) towards a regime shift characterised by a eutrophic state (turbid water) passing through three stages (a) – (c). We propose that a negative correlation of nestedness temperature to biodiversity arises in stage (b) because rising stress on the community (phosphorus influx in this case) causes species accumulation rather than turnover with the loss of canaries and consolidation of strong competitors. During and after the regime shift (stage (c)), both the biodiversity and the nestedness temperature decrease and the system becomes a compact, nested community with very few species left. This produces a positive correlation.

For a potentially broad range of ecosystems that are characterised as having well-mixed or homogeneous environmental variables, we propose three key stages towards a regime shift (see Fig. 3).

- (a) Long before the regime shift: Temperature and biodiversity both fluctuate without obvious trend in biodiversity and temperature. However there is a weakly positive correlation (i.e., more biodiversity correlates with more disorder). This represents a healthy community that adds and loses canary species more or less at random.
- (b) Prior to the regime shift up to its cusp: Temperature and biodiversity fluctuate without obvious trends in either, but now with a strongly negative correlation (i.e., more biodiversity correlates with less disorder). Upward fluctuations of biodiversity add competitively dominant (keystone) and fast-fugitive (weedy) species at the expense of canaries, thereby also raising order. In effect, the out-of-phase fluctuations of biodiversity and temperature ratchet out the canaries and replace them with more strongly connected species, reflecting a tightening of the community in response to stress.
- (c) During and after the regime shift: Temperature falls dramatically with biodiversity in a strongly positive correlation, as keystone species are lost, leaving only the few most abundant and robust (weedy) species that have always been present.

These hypotheses and assumptions are consistent with empirical results obtained from Lake Erhai, Yunnan Province of China. A dataset for changes in a diatom community over time was obtained from a 63-cm sediment core in Lake Erhai, representing about 500 years of sedimentation prior to a critical transition in 2001, and 8 years post-transition. We developed a temporal analysis of the diatom community to obtain information about the community composition and structure before and after a critical transition that took place in 2001, according to results found by (Wang et al. 2012).

Before the critical transition, the community shows relatively high temperature (low levels of nestedness), though decreasing gradually towards the tipping point. During and after the critical transition, temperature decreases drastically. We correlated nestedness temperature with community biodiversity, finding that the sign of the correlation switched from positive to negative at about 50 years prior to the tipping point, and then back to positive immediately after the tipping point. We interpret these changes as a potential signal of ecosystem stress, which leads to the community tightening up as it loses first the canary species and finally the keystone species as it goes over the critical transition.

These empirical results provide a benchmark for the generation and analysis of the artificial ecosystems presented below. With limited capacity for further progress through analyses of empirical datasets, because of the financial and time costs involved in obtaining them, we perform numerical simulations of communities similar to the ones found empirically, and we test our assumptions on them. Such

simulations can be understood as an initial evaluation of our hypothesis. A next step would be the development of more detailed agent based models that would explicitly capture the important processes that we identified in (a), (b) and (c) above.

## Methods

### Simulation experiments

We generated artificial ecosystem matrices, that simulate community distributions of species under the influence of different types of stressors. The objectives of these simulations are: (1) to test for variations in the nestedness temperature in response to stressors, and (2) to identify relationships between nestedness and biodiversity through time. We anticipate that the results of these numerical models may point towards a robust early warning signal for regime shifts in community composition.

### Calculating nestedness

Nestedness temperature measures how much the incidence matrix departs from perfect nestedness. To calculate this metric, the list of species present in a series of times is summarised in an incidence matrix of presence-absence. The rows and columns of the incidence matrix are reordered so that nestedness is maximised, using the algorithm proposed by Rodríguez-Gironés and Santamaría (2006). The matrix is re-arranged to show species presences on the top left corner and the absences away from the top left corner. This creates a matrix where the columns rank species rarity (increasing from left to right) and the rows rank species richness (increasing bottom to top). Then an isocline of perfect nestedness is calculated to show the expected distribution of presences if the matrix were perfectly nested.

Absences to the top and left of the isocline are defined as unexpected, and so are presences below and to the right (see Fig. 4). The matrix nestedness temperature is calculated as the sum of squared deviations from the isocline of unexpected presences and absences divided by the maximum value possible for the matrix, multiplied by 100. Thus, the temperature is a non-dimensional index measuring how much the matrix departs from the perfectly nested state (Rodríguez-Gironés and Santamaría 2006; Almeida-Neto et al. 2007).

### Generating power-law population distributions

Many populations follow power-law frequency distributions with respect to their abundance across time (Mitzmacher 2004; Allen et al. 2001). Accordingly, we generated power-law frequency distributions of theoretical communities which we used as the basis of numerical experiments. We generated a matrix with 150 rows and 150 columns, where the  $M(i, j)$  represents the species in column  $j$  at time  $i$ . Each row is an independently generated power-law community distribution of 150 species.

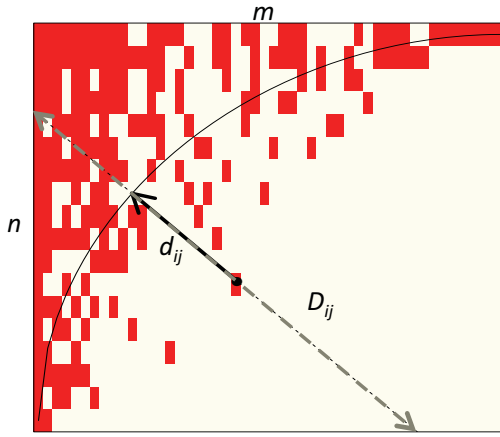


Figure 4: Incidence matrix of  $m$  species identities (ranked from most to least often present), in  $n$  samples (ranked from most to least species rich, bottom to top). The isocline curve corresponds to the theoretically perfect nested structure of the system. All the red squares are presences and white squares denote absences. Temperature is calculated from the sum of all squared deviations of unexpected presences/absences  $u_{ij} = (d_{ij}/D_{ij})^2$ . Modified from Oksanen 2012.

Once a core matrix was generated, we applied a number of forcing functions to it. By doing this we introduced what would be analogous to stressors of change in the environment (e.g. changes in nutrients, salinity, pH). The proposed functions had an immediate effect in the change of nestedness temperature across the nested blocks over the temporal scale. Equally, these changes in the community structure, affected Hill's biodiversity index. The main objective of these numerical simulations was to obtain correlations between these quantities to compare to the empirical data on diatoms from Lake Erhai.

### Core matrix analysis

Using the method proposed in the previous section, we generated 100 matrices with power law distributions. Each core matrix was transformed into a presence-absence matrix from which we found an average nestedness temperature of  $T \approx 87.57$ . Figure 5 shows the nestedness temperature for a matrix with these characteristics. The high temperature of the community matrix does not show any type of structure. However, as we will see in the next section, when the system was subjected to external perturbations, the community acquired structure and the nestedness temperature decreased.

### Forcing functions

We have limited knowledge of the type of stressors acting on a real ecosystem. Consequently, we formulated a series of forcing functions that reduce the number of species in the community proportional to the value of parameter,  $z$ . The different forcing functions affect the abundance of species in

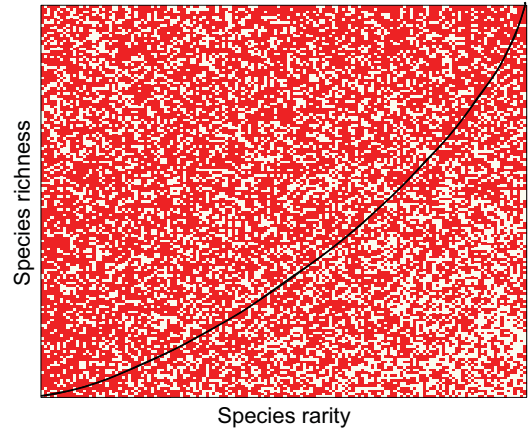


Figure 5: An example of a high incidence temperature system that shows little structure. A dataset with 150 species following a power law distribution for 150 times depicts an incidence temperature of 87.5. The continuous black line represents the isocline for a perfectly nested structure. The empty squares above the isocline denote surprising absences, while the red squares below the isocline denote surprising presences.

different ways. For example one forcing function will preferentially remove species with high abundance, another will reduce species with low abundance. We are motivated to do this as we seek to understand how the correlation between nestedness temperature and biodiversity changes under different perturbations and stressors.

**Max** Eliminates the species with the highest number of individuals in each row:  $f(z) = M_{i,K} - \frac{\max(M_{i,K})}{2} - z$ . These species are located in the peak of the distribution, and must exceed  $\frac{\max(M_{i,K})}{2}$  because the number of species with such high abundances is very small.

**Min** Eliminates the species with the lowest number of individuals in each row:  $f(z) = M_{i,K} - \min(M_{i,K}) + z$ . These species are located in the tail of the distribution and comprise a large percentage of the community.

**Middle** Eliminates the most abundant and the least abundant species, retaining those in the middle of the distribution:  $f(z) = \min(M_{i,k}) + z < M < \frac{\max(M_{i,K})}{2} - z$ .

**Outflux** Decreases the number of individuals of each cell in the matrix by a percentage  $z$  given by a fixed parameter  $f(z) = M_{ij} - z$ .

**In-Out** This function simulates the influx and outflow of individuals  $f(z) = M_{ij} - z + i$ . The population in each cell increases with time (row number  $i$ ) and decreases with the size of  $z$ .

Here we have assumed that  $M_{i,K}$  corresponds to the  $i$ th row for all the columns of the matrix  $M$ ,  $M_{ij}$  correspond

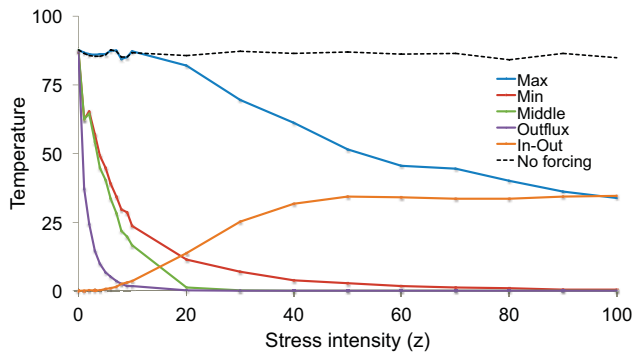


Figure 6: Changes in nestedness temperature of an artificial community of 150 species, over 150 time steps in response to increasing intensity ( $z$ ) of each of six stressors. The stressors that eliminate the rare species first (e.g., **Min-Outflux**) tend to reduce nestedness temperature most rapidly, reflecting an increased ordering and predictability in species composition. Function **In-Out** simulates a continuous flux of species through the community, which raises nestedness temperature from zero, reflecting the reduced predictability of composition induced by the turnover of species.

to the cell  $(i, j)$  of the matrix and  $z$  is a scalar parameter of fixed value. These functions are not an exhaustive selection; they give general indications of what responses stressors might induce in an originally well-mixed community.

In order to calculate how nestedness temperature changed over time in response to the different forcing functions, each matrix was divided in 135 consecutive nested subsets, where each subset had 15 rows (i.e. rows 1 to 15, then rows 2-16, 3-17, and so on). The nestedness temperature was then obtained for each subset.

### Results

The nestedness temperature varied with respect to the forcing function and the size of the forcing parameter  $z$  (see Fig. 6). In the cases of functions **Max**, **Min**, **Middle** and **Outflux**, the temperature decreased as the intensity of the forcing function increased (as  $z$  increases), with differently decreasing curves depending on the forcing function. This is expected since as the number of species decreases losing the canary species, the community becomes more nested. In the case of function **In-Out**, we observed that the nestedness temperature increased from 0 to 34.57. The reason for this is that the competing in- and out- fluxes balance each other as the parameter  $z$  increases. When  $z = 0$  the population in each row increases with the row number, e.g. each species in row  $i$  will increase in  $i\%$  adding more individuals to the community. As  $z$  increases, individuals leave the row community  $i$ , balancing out the fluxes into and out of the community. Consequently, when  $z \geq 100$ , the population only decreases until it eventually vanishes.

As a preliminary analysis, we choose three matrices from the whole set displayed in Fig. 6 and investigate the possi-

ble correlations between the community structure and biodiversity. We analysed relatively nested communities with nestedness temperature of about 25 units for three forcing functions in order to determine how a drop/increase in temperature and Hill's biodiversity changes the correlation between these two quantities. The results for these analyses are summarised in Table 1.

We chose functions **Min** and **Middle** since they best account for a linear loss of canary species, while **In-Out** shows drastic changes in nestedness temperature. Our analysis shows that a linear increase or decrease in nestedness temperature maintains the sign of the correlation between temperature and biodiversity. This implies that changes in community structure will maintain the same type of dynamics on the biodiversity. Conversely, we found that a sharp decrease or increase in nestedness temperature changes the correlation sign. The best example for this change of sign in the correlation is given by function **In-Out** displayed in Fig. 7.

Function	Rows	T	H	$r$	$P$
Min	20 – 75	+	n.t.	+0.228	0.0940
	80 – 110	–	n.t.	+0.287	0.1229
Middle	2 – 82	n.t.	n.t.	+0.164	0.1444
	82 – 112	–	n.t.	+0.320	0.0897
In-Out	1 – 7	n.t.	+	–0.620	0.0657
	8 – 20	++	++	+0.856	0.00001
	21 – 32	--	--	+0.428	0.0016
	33 – 60	n.t.	–	+0.665	0.0425

Table 1: Trends in nestedness temperature (T) and Hill's biodiversity (H) and their correlation coefficient ( $r$  and significance  $P$ ). Trends are + or ++ for weakly or strongly positive, – or -- for weakly or strongly negative, n.t. for no trend. Correlations between nestedness temperature and Hill's biodiversity index show that linear changes maintain the sign of the correlation, while drastic changes result in a correlation shift.

A sharp change of temperature and Hill's diversity index results in a change of correlation sign from negative (while the biodiversity starts increasing the nestedness remains more or less constant due to the species flow into and out of the system) to positive (a sharp increase in biodiversity results in a higher temperature due to the large influx of individuals into the community), then a sharp drop in the temperature and Hill's diversity index results in a positive correlation (as biodiversity decreases sharply the temperature decreases as well as only few structured species remain). Notice that after this decrease in temperature and biodiversity, the correlation remains positive. This is due to the continued decrease of biodiversity and temperature after the shift. Notice also that, when the drop of the temperature is abrupt, the correlation is stronger.



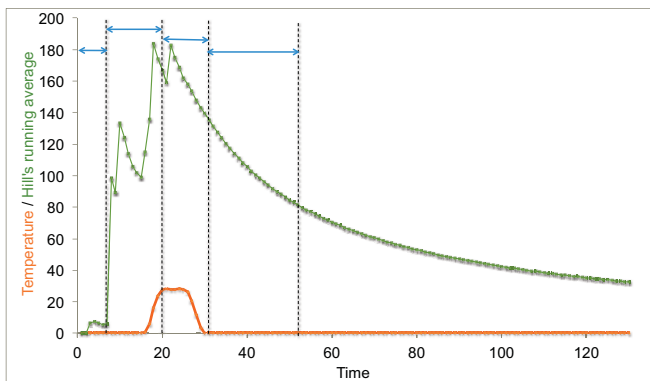


Figure 7: Nestedness temperature and running average of Hill's diversity index  $N_2$  for the numerically generated matrix **In-Out** with a stress intensity of  $z = 30$ . Both, the temperature and the Hill's index increases and decreases very drastically between row 8 and row 30. The arrows between vertical dotted lines show the regions analysed to obtain the correlation between these two non-smoothed quantities. These correlations are summarised in Table 1.

## Discussion

Our study of the biodiversity and nestedness of an artificial ecosystem aims to identify sensitive detectors of changes in environmental processes that drive regime shifts. Our hypothesis assumes that “canary species” (those most susceptible to extinction) function as detectors of drivers of change, preceding the approach to a critical tipping point.

We generated artificial populations with power law distributions and subjected them to a number of drivers of change. We correlated nestedness with biodiversity and established which conditions of external stress produced which correlations. We found that large changes in biodiversity and nestedness can lead to significant changes in the sign and magnitude of the correlation of the two measures. Such a change is qualitatively similar to that observed in a real-world lake ecosystem. We hypothesise that the change in correlations in the lake system are produced by the loss of ecosystem canary species in response to external perturbations. As these vulnerable species are lost the community ‘tightens’. Increasing the intensity of perturbation would lead to further changes such that keystone species were affected. This would not only change the correlation between biodiversity and nestedness, but also lead to an imminent collapse in the systems as major ecosystem level properties and structure would rest.

Our analysis presented here was performed for a small number of systems. In order to further explore the utility of an early warning signal of an impending regime shift that is generated by the correlation of biodiversity and nestedness we propose the development of a series of agent based models that would build on this initial work. These models would allow the inclusion of traits such as growth rate

and competitive ability. In doing so we would be able to explore the interactions between competitively dominant but slow growing keystone species, fast growing but competitively weak species and canary species that are both poor competitors and slow growing. If our initial assumptions are correct, then what would emerge in response to progressive perturbations on the population would be a robust indicator of a regime shift.

## Acknowledgements

This work was supported by EPSRC Bridging-the-Gap grant EP/K503575/1.

## References

- Allen, A. P., Li, B.-L., and Charnov, E. L. (2001). Population fluctuations, power laws and mixtures of lognormal distributions. *Ecology Letters*, 4(1):1–3.
- Almeida-Neto, M., R Guimarães Jr, P., and M Lewinsohn, T. (2007). On nestedness analyses: rethinking matrix temperature and anti-nestedness. *Oikos*, 116(4):716–722.
- Almeida-Neto, M. and Ulrich, W. (2011). A straightforward computational approach for measuring nestedness using quantitative matrices. *Environmental Modelling & Software*, 26(2):173–178.
- Atmar, W. and Patterson, B. D. (1993). The measure of order and disorder in the distribution of species in fragmented habitat. *Oecologia*, 96(3):373–382.
- Carpenter, S. and Brock, W. (2006). Rising variance: a leading indicator of ecological transition. *Ecology letters*, 9(3):311–318.
- Cutler, A. (1991). Nested faunas and extinction in fragmented habitats. *Conservation Biology*, 5(4):496–504.
- Elmendorf, S. C. and Harrison, S. P. (2009). Temporal variability and nestedness in california grassland species composition. *Ecology*, 90(6):1492–1497.
- Fischer, J. and Lindenmayer, D. B. (2005). Nestedness in fragmented landscapes: a case study on birds, arboreal marsupials and lizards. *Journal of Biogeography*, 32(10):1737–1750.
- Guttal, V. and Jayaprakash, C. (2008). Changing skewness: an early warning signal of regime shifts in ecosystems. *Ecology Letters*, 11(5):450–460.
- Hastings, A. and Wysham, D. B. (2010). Regime shifts in ecological systems can occur with no warning. *Ecology Letters*, 13(4):464–472.
- Heino, J., Mykrä, H., and Muotka, T. (2009). Temporal variability of nestedness and idiosyncratic species in stream insect assemblages. *Diversity and Distributions*, 15(2):198–206.
- Hill, M. O. (1973). Diversity and evenness: a unifying notation and its consequences. *Ecology*, 54(2):427–432.
- Jones, C. G., Lawton, J. H., and Shachak, M. (1994). Organisms as ecosystem engineers. *Oikos*, 69:373–386.

- Krebs, V. (2012). Community networks. <http://www.thenetworkthinkers.com>.
- Lenton, T., Livina, V., Dakos, V., Van Nes, E., and Scheffer, M. (2012). Early warning of climate tipping points from critical slowing down: comparing methods to improve robustness. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1662):1185–1204.
- Magurran, A. E. and Magurran, A. E. (1988). *Ecological diversity and its measurement*, volume 168. Princeton university press Princeton.
- May, R. M. (1973). *Complexity and stability in model ecosystems*. Princeton University Press, New Jersey.
- McCann, K. S. (2000). The diversity–stability debate. *Nature*, 405(6783):228–233.
- Mitzenmacher, M. (2004). A brief history of generative models for power law and lognormal distributions. *Internet mathematics*, 1(2):226–251.
- Montoya, J. M., Pimm, S. L., and Solé, R. V. (2006). Ecological networks and their fragility. *Nature*, 442(7100):259–264.
- Oksanen, J. (2012). Design decisions and implementation details in vegan. <http://mirrors.dotsrc.org/pub/pub/mirrors/cran/web/packages/vegan/vignettes/decision-vegan.pdf>.
- Patterson, B. D. and Atmar, W. (1986). Nested subsets and the structure of insular mammalian faunas and archipelagos. *Biological Journal of the Linnean Society*, 28(1-2):65–82.
- Rodríguez-Gironés, M. A. and Santamaría, L. (2006). A new algorithm to calculate the nestedness temperature of presence–absence matrices. *Journal of Biogeography*, 33(5):924–935.
- Scheffer, M., Bascompte, J., Brock, W. A., Brovkin, V., Carpenter, S. R., Dakos, V., Held, H., Van Nes, E. H., Rietkerk, M., and Sugihara, G. (2009). Early-warning signals for critical transitions. *Nature*, 461(7260):53–59.
- Scheffer, M., Carpenter, S., Foley, J. A., Folke, C., and Walker, B. (2001). Catastrophic shifts in ecosystems. *Nature*, 413(6856):591–596.
- Scheffer, M., Carpenter, S. R., Lenton, T. M., Bascompte, J., Brock, W., Dakos, V., van de Koppel, J., van de Leemput, I. A., Levin, S. A., van Nes, E. H., et al. (2012). Anticipating critical transitions. *Science*, 338(6105):344–348.
- Scheffer, M. and Nes, E. H. (2007). Shallow lakes theory revisited: various alternative regimes driven by climate, nutrients, depth and lake size. *Shallow Lakes in a Changing World*, pages 455–466.
- Simpson, E. H. (1949). Measurement of diversity. *Nature*, 163(4148):688.
- Strogatz, S. H. (2001). Exploring complex networks. *Nature*, 410(6825):268–276.
- Timi, J. T. and Poulin, R. (2008). Different methods, different results: temporal trends in the study of nested subset patterns in parasite communities. *Parasitology*, 135(1):131.
- Ulrich, W., Almeida-Neto, M., and Gotelli, N. J. (2008). A consumer’s guide to nestedness analysis. *Oikos*, 118(1):3–17.
- Wang, R., Dearing, J. A., Langdon, P. G., Zhang, E., Yang, X., Dakos, V., and Scheffer, M. (2012). Flickering gives early warning signals of a critical transition to a eutrophic lake state. *Nature*, 492(7429):419–422.

# SimianWorld - A Study of Social Organisation Using an Artificial Life Model

Sue Attwood, Lola Cañamero and Rene te Boekhorst

Adaptive Systems Research Group, University of Hertfordshire,  
College Lane, Hatfield, Hertfordshire, AL10 9AB

s.c.attwood@herts.ac.uk, l.canamero@herts.ac.uk, r.teboekhorst@herts.ac.uk

## Abstract

In studies of social behaviour it is commonly assumed that individual complexity is the origin of intricate social interactions. In primates for example, social complexity is attributed to their intelligence and it is argued by many that the cognitive capacity of primates are especially manifest in the way they regulate their social relationships. Whereas the complex societies of non-human primates are considered to be as a direct result of their cognitive abilities this assumption is not made about social insects. In the absence of certain cognitive abilities their complex societies and structurally sophisticated nests are thought to arise from self-organisation. Since it is unlikely that cognitive capacities are all-or-nothing, usually integrating a range of mechanisms, it is possible that different species use similar cognitive mechanisms resulting in different behavioural outcomes.

## Introduction

When observing and analysing complex systems, such as social structure, scientists tend to use separate explanations for each observed part or component of that system. Furthermore, they tend to explain the cause of complex patterns of behaviour observed at a higher lever (such as the relationship or the group) as if it resides within individuals, Hemelrijk (2004). In studies of social behaviour it is commonly assumed that individual complexity is the origin of intricate social interactions. In primates for example, social complexity is attributed to their intelligence and it is argued by many that the cognitive capacity of primates are especially manifest in the way they regulate their social relationships, Tomasello and Call (1997). In contrast, the complex organisations of social insects with large colonies and structurally sophisticated nests (honey bees for example) are not attributed to intelligence, their cognitive abilities are known to be limited and so any complex traits displayed are thought to arise from self-organisation, Hemelrijk and Puga-Gonzalez (2012). Explaining aspects of behaviour as different intelligent or rational decisions, leads scientists to come up with theories to integrate these separate aspects. The problem with such theories is that they tend to be complicated and assumption loaded. There are

considerably more possible reasons why cognitive capacity might not be found than there is evidence of their existence. Until such evidence is found the most parsimonious assumption should be that species, closely related or not, that show similar solutions to similar problems are likely to use similar cognitive mechanisms. It is therefore, theoretically possible that different species achieve similar outcomes in different ways and that unique outcomes do not always mean unique processes, de Waal and Ferrari (2010).

This paper describes an individual-orientated, agent-based model called *SimianWorld*. *SimianWorld* can be reasonably called an example of Artificial Life, *Artificial* because it is not designed to be in close correspondence with any previously observed life forms; *Life* because it expands our observable universe with entities who “live lives” in which we can observe patterns normally pre-eminently associated with real life, Hogeweg (1988).

As is typical for Artificial Life, the present study uses a bottom-up approach to generate hypotheses and alternative explanations for complex behaviour and social relationships in terms of simple behavioural rules, limited cognitive assumptions and environmental structure.

*SimianWorld* incorporates a simplified version of some aspects of animal behaviour and represents a kind of caricature. The advantage of a caricature is that by exaggerating patterns they become more visible, Hemelrijk (1999). In contrast to the naturally incomplete explanation of animal behaviour, *SimianWorld*'s complete description allows us to establish what factors and dynamics are responsible for the emergent social patterns. If patterns of behaviour happen to correspond to those observed natural systems new hypotheses for existing explanations may be derived from the model. Such hypotheses are often counter-intuitive and innovative.

## Non-Human Primates as Study Objects

To understand the role of positive and negative affect-based interactions in the formation of social relationships as a response to internal state and environmental conditions, non-human primates are good study objects. This is because the social organisation of many nonhuman primate species

relatively simple compared to that of humans but complex enough to be interesting. Primates have varied and diverse ways of expressing themselves socially and form various kinds of relationships. Furthermore, studies of non-human primates have illustrated the link between negative and positive affect-based interactions and social structure by relating dominance hierarchy to despotic and egalitarian societies.

From the observation that most primates live in groups, the rational deduction is that sociality must be a beneficial trait. The main advantage of group life is assumed to be protection, either against predators, van Schaik (1983) or against conspecific rival groups, Wrangham (1987). Competition plays an important but different role depending on your viewpoint. To van Schaik, competition for food with other group members is an inevitable consequence of group life and this, together with its benefits, determines optimal group size. Wrangham (1980; 1987), in contrast, sees competition between groups as the ultimate cause of sociality, in that large groups can displace smaller ones from vital resources. van Schaik combined within-group and between-group competition in an extended model, van Schaik (1989), to arrive at more detailed predictions on social relationships. He also added two different types of competition, contest competition, which occurs when individuals compete directly over resources, and scramble competition, which is based on the assumption that individuals lose resources because other group members have already used them. From the resulting combinations, van Schaik drew up a classification of primate social systems into competitive regimes. The matching types of social organisations in terms of “despotic” or “egalitarian” societies, Vehrencamp (1983), are then interpreted as predictions. For example, if ecological conditions lead to contest competition between groups, the formation of alliances will be important and therefore dominants must relax contest competition within the group. Otherwise subordinates might either refrain from taking any risks in between-group contests or even defect to another group, van Hooff and van Schaik (1992). Thierry et al. (2004) suggests that macaques are a particularly good model genus for studying the above model of primate social organisation. There are approximately 21 genetically closely related macaque species that can be classified both by diet (type of food available), distribution of food and by social relationships (despotic or egalitarian), Thierry (2006), van Hooff and van Schaik (1992).

### **Artificial Life Models of Primate Social Organisation**

te Boekhorst and Hogeweg (1994) used individual oriented models of “artificial apes” to study the formation of social groups based on simple gender differences in looking for food (by females) or females (by males). Limited manipulations of the rules and the environmental conditions lead to the emergence of chimpanzee-like or orangutan-like group

structures. The model thus arrived at alternative explanations for observed social structure without the need of either neo-darwinist assumptions about kinship selection, reciprocal altruism and optimal foraging theory or conjectures about sophisticated cognitive capacities. Hemelrijk (1999), has shown that dominance hierarchy, spatial social structure (with dominants in the centre of the group and subordinates at the periphery) and social organisation (despotic versus egalitarian societies) all emerged in an artificial world in which the behaviour of the agents was steered by only a few basic social rules (if lonely approach others, if others are too nearby either flee or chase them away), limited cognitive capacity (chase away another agent if its perceived dominance is lower than that of your own) and simple social dynamics (dominance of an individual increases especially if it wins by accident against expectation). The model also resulted in similarities and new insights about the relationship between aggressive behaviour, dominance and social structure as found in the macaque species studied by Thierry.

In summary, the above mentioned Artificial Life models studied either the effect of food type and food distribution on the composition of groups, but did not address social relationships, te Boekhorst and Hogeweg (1994) or they studied the emergence of social relationships and structures based on dominance interactions but without modelling food resources, Hemelrijk (1999). As such they incompletely addressed the framework of van Schaik (1989). One of the objectives of this research is to combine the social aspects of the Hemelrijk model with ecological resources modelled by te Boekhorst and Hogeweg (1994).

### **Higher-levels of Affective States**

The above mentioned Artificial Life models adopt a very simple implementation of affective states and affect-based interactions (mainly leading to group members moving away or staying close to other agents). However, in real life affective states are more complex and include for example emotions, drives, pleasure, pain, attitudes, moods and values. At this point it is important to distinguish between motivations and emotions. Motivational states such as hunger and thirst, for example, are drives that constitute urges to action based on internal needs related to survival and are seen as homeostatic processes which maintain a controlled physiological variable with a certain range. Emotions however, can be regarded as second-order modifiers or amplifiers of motivation, Cañamero (1997). Neurobiology attempts to characterise emotions as complex reflexes that regulate control mechanisms to excite or inhibit response to stimuli - both internal and external.

Motivational and emotional mechanisms have been attributed with complementary roles - while motivation is concerned with the operations of appetitive processes that try to activate action as a response to deprivation, emotion is derived from processes that try to stop ongoing behaviour



i.e. it is concerned with satiation and equilibrium, Cañamero (1997). Disciplines as diverse as psychology, neurobiology, and philosophy have studied the nature of emotions. These diverse disciplines have focused on different aspects of emotional phenomena, sometimes proposing incompatible theories about them. However, they all share the same underlying idea: that emotions, whatever they are, have an adaptive value – they serve a purpose. This hypothesis is known as the functional view of emotions, Cañamero (2001). Attempts to integrate higher level affective states into agent architectures has been receiving increasing attention in Artificial Life and Cognitive Science. Proposed reasons for this, Cañamero (2001), are that affective agent-based systems can be used in the following ways:

1. As a test bed for theories about affective states in animals and humans, a synthetic approach that complements analytical studies of natural systems.
2. To explore the role that affective states plays in biology in order to develop and exploit mechanisms that ground and enhance autonomy, adaptation and social interactions.

### Implementation of Affective States in the SimianWorld Model

The *SimianWorld* agent architecture consists of motivational states and behaviours. There are two homeostatic variables that correspond to the internal resources of energy and sociability. Together these variables represent the internal state of an agent and each has an “ideal state” (i.e. a set point or norm value). The degree to which each variable deviates (the error) from this ideal state constitutes internal stimuli and directly influences agent behaviour. The agent’s motivational states are abstract representations of a propensity to behave in a particular way as a result of a combination of internal and external stimuli.

### The Agent Architecture

External stimuli comes in two forms, the first of which is clumps of food. By eating the food agents modify both the external environment and their own internal level of energy. The other type of stimulus is the agents themselves. The presence of other agents stimulates social behaviours (like groom and attack for example) that modify the agent’s internal level of sociability.

So, the motivational state is the mechanism that corrects the level of error in the homeostatic variables representing physiology through the execution of appropriate behaviours. How the agents react in social situations such as eating or resting for example, and how they interact, will largely depend on the distribution pattern of food (the amount, distribution and renewal rate of food can be closely controlled). Interactions can be of either a positive nature, like grooming for instance or be negative in nature, as in the case of an

attack. When calculating motivation intensity and thereby activating a behaviour it is fundamental to define a rule combining external and internal stimulus. *SimianWorld* uses the following method:

1. The motivation  $j$  is calculated in the following way: For each motivation  $j$ :
  - i Calculate the intensity of the motivation’s drive, proportional to the error of its homeostatic variable  $v(e_{v_j} \in [0, 1])$
  - ii Calculate the effect of the presence of external stimuli  $k$  influencing the intensity of the motivation  $j$  (an increased incentive):  $s_{k_j} \in [0, 1]$
  - iii  $m_j = e_{v_j} + (s_{k_j} \times e_{v_j})$  is the final intensity of  $j$

The motivation with the highest intensity is selected and an appropriate behaviour activated.

2. The error is calculated in the following way:
  - i Calculate the distance between the set point and the limit:  $ld_v = abs(l_v - p_v)$
  - ii Calculated the distance between the actual variable and the set point:  $vd_v = abs(v_v - p_v)$
  - iii Calculate the normalised error

$$e_v = \begin{cases} vd_v/ld_v & \text{if } ld_v > abs(l_v - v_v) \\ 0 & \text{otherwise} \end{cases}$$

The way in which motivation and behavioural intensity are combined is a problem that has been extensively researched by ethologists. *citetA24* demonstrated that a simple adaptive rule,  $Cue \times Deficit$ :  $m_j = s_{k_j} \times e_{v_j}$  gives the behaviour of an agent both opportunism and persistence.

The problem with this rule is the lack of motivational arousal in the absence of external stimuli. This model would prove fatal in models in which the motivational state leads to the selection of appetitive<sup>1</sup> behaviours.

Therefore *SimianWorld* uses the following extension to the formula proposed by Tyrrell (1993):

$$Deficit + Cue \times Deficit: m_j = e_{v_j} + (s_{k_j} \times e_{v_j})$$

<sup>1</sup>Animal searching behaviour. The variable introductory phase of an instinctive behaviour pattern or sequence, e.g., looking for food or looking for others.

Behaviour	Homeostatic Variables	
	Energy	Sociability
Eat	↑	with others ↑ alone ↓
Groom (source)	↑	↑
Groom (target)	↓	↑
Attack (source)	↓	↓
Attack (target)	↓	↓
Find Others	↓	↓
Find Food	↓	↓
Wander	↓	↓
Avoid	↓	↓
Rest	↓	with others ↑ alone ↓

Table 1: Behaviours and their Effect on the Homeostatic Variables

Using the above method agents select a behaviour most likely to satisfy their immediate needs. For instance an agent whose energy level is satisfied (above the set point or norm value) but whose level of sociability is low may seek other agents (Find Others). However, finding others uses energy and at some point the agent may need to alter its behaviour. Every agent's levels of energy and sociability are calculated at each time tick, time ticks are regulated using the GetTick-Count function in C++ which returns the system time in milliseconds. The sequence in which each agent is scheduled to activate its selected behaviour is reordered at each time tick.

Agents in *SimianWorld* are “individuals” possessing a behavioural repertoire that includes sensing their local environment, changing the environment and changing their own position in the environment. Each agent is initialised with the set point values of energy and sociability, in addition they are given a random starting position on the grid and a random direction. The grid squares that immediately surround each agent are designated as personal space with a further number of grid squares designated as their vision range, see Figure 1. The behaviours modelled by *SimianWorld* are: eat, attack, groom, wander, rest, find food, find others and avoid others, see Table 1.

### The System Architecture

The system architecture has two elements; the agents and a simple ecology. The agents are homogeneous and live in a flat, boundary restricted world, their territory. The landscape has no features apart from clumps of food. The size of the territory can be altered but for the all the simulations



Figure 1: Snapshot of agents in *SimianWorld*

run to date this has been set to the maximum size of 130 x 115 (14950 grid locations). Agents can move in any direction – randomly chosen when the boundary is reached. The amount and distribution of food can also be closely controlled through the interface, as can the population size and vision range of the agents. There is a full graphical display of the simulated environment that shows the agents (displaying their field of vision is optional but is costly in terms of visual rendering) position within the grid and the location of food clumps, see Figure 2.



Figure 2: The Graphical Interface

### Experimental setup

As a first experiment, we investigated the possible relationships between on the one hand measures of distribution and amount of food and on the other hand the reciprocation of dominance interactions (as a reflection of negative affect) and friendship interactions (as a reflection of positive affect). Reciprocation was crudely measured as the correlation coefficient between row and column totals of “dominance” and

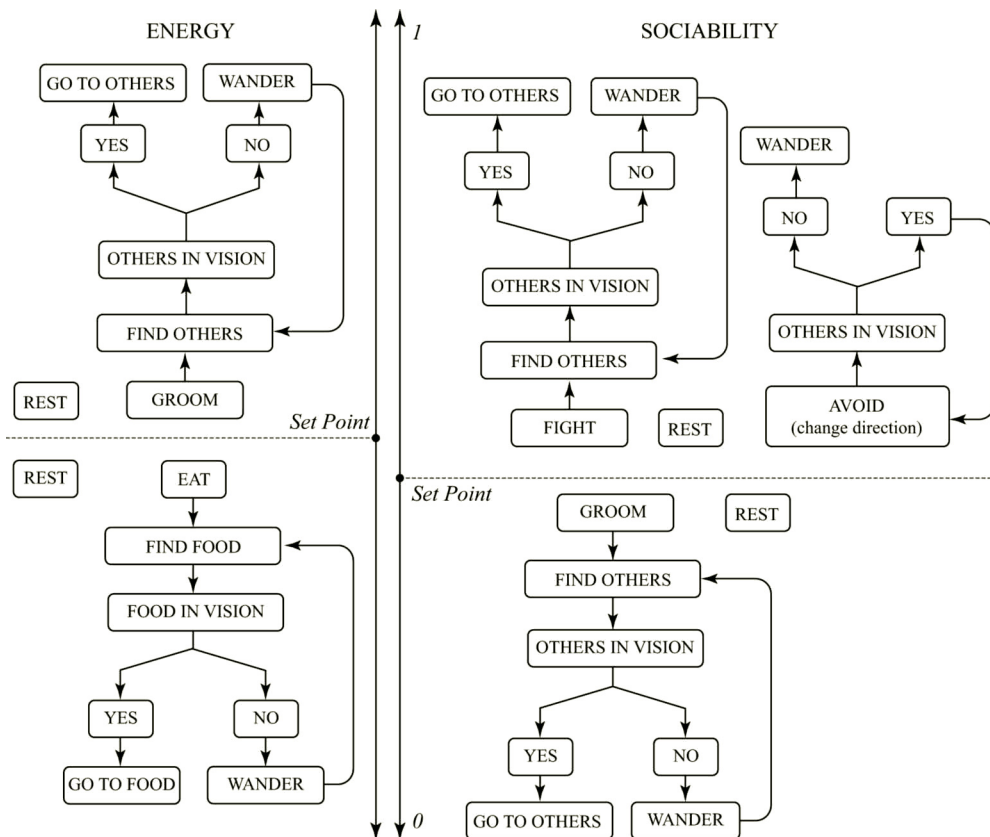


Figure 3: The SimianWorld Architecture

“friendship” matrices. These correlation coefficients were calculated for the following environmental conditions:

1. A large amount of food highly clumped
2. A small amount of food highly clumped
3. A large amount of food widely distributed
4. A small amount of food widely distributed

The distribution of food in clumps is calculated in a very simple way, the amount of food (set to between 1 and 500 grid squares occupied by food) is divided by the distribution level of food (set to between 1 and 25). Thus, 400 grid squares of food, divided by a distribution factor of 5 produces 5 clumps of food in 80 grid squares. At the start of a simulation, each clump begins at a randomly chosen grid square about which the required number of grid squares of food are closely grouped. Food is “consumed” by agents and can be renewed, rejoining a previously established clump at set time intervals (ticks). All the above mentioned variables are set at the start of a simulation run through an Options panel in the interface. The renewal rate of food is dependent

on each combination in order to maintain initial conditions where possible.

For each of the 4 environmental conditions 5 runs were conducted with a population of 10 agents and 5 runs with 20 agents (40 runs in total). Each run consisted of 26010 time ticks and at every time tick each agent’s internal state was assessed and an appropriate behaviour executed. During each run and at every time tick the agent’s spatial position, behaviour, levels of sociability and energy were recorded. In addition, at intervals of 600 time ticks during the run a log was taken of all the positive and negative interactions and these were stored in the form of a matrix.

Each time two agents meet, their interactions were characterised as either negative or positive and these outcomes are summarised respectively in a “dominance” and a “friendship” matrix. The values of the cells in the matrix, reading from left to right, are the frequencies by which a row-labeled agent initiated an interaction with a column-labeled agent. Reading from top to bottom, the values are the frequencies by which a column-labeled agent received an interaction from row-labeled agents.

Accordingly, row-totals are the total frequencies agents

started interactions irrespective the identity of the receivers, and column-totals are the overall count of interactions agents received (summed over all the initiating agents). In this sense a positive correlation between row and column totals implies that agents that initiated more interactions also received more of those interactions and thus functions as a crude measure for reciprocity.

## Results

Although there were four different environmental conditions each tested with both large and small populations only some produced results that could be considered interesting. Interesting in the sense that the emergent social patterns indicated either despotic or egalitarian social structures.

1. With a large amount of food and a small population (10), negative reciprocation of dominance interactions implies agents initiate a greater number of “aggressive” interactions than they receive. Thus indicating a more “despotic” social structure. See Figure 4.
2. With large clumps of food and a large population (20) there is a negative reciprocation of dominance interactions. This implies that under these conditions agents *tend* to initiate more “aggressive” interactions than they receive. Indicating a more “despotic” social structure (results are not significant  $p = 0.06$ ). See Figure 5.
3. With a low amount of food and a large population (20), negative reciprocation of friendship interactions implies agents tend to initiate a greater number of “friendly” interactions than they receive (results are not significant  $p = 0.09$ ). See Figure 6.

## Conclusion

Models of Artificial Life have in the past been used to study the effect of food type and food distribution on the composition (fission/fusion) of groups, te Boekhorst and Hogeweg (1994), but did not include any details of social relationships. They have also been used to study the emergence of social relationships and structures based on dominance interactions but without modelling food conditions, Hemelrijk (1999). The main objective of this research is to combine aspects of social relationships and ecological resources in one model and to study the structure of the societies that emerge.

One specific aim of this research is to relate the results of future tests to that of social structures found in nature (macaques).

Work to date has concentrated on building an individual-orientated, agent-based software prototype (Simian World) that uses an Artificial Life approach that incorporates a simplified version of some aspects of animal behaviour. In contrast to the naturally incomplete explanations of animal behaviour, the model's complete description allows us to estab-

	Pearson Correlation Coefficients	
	low amount	high amount
low clumping	-0.546	-0.395
	0.014	-0.266
	0.102	-0.566
	-0.246	-0.648
high clumping	-0.362	-0.694
	-0.608	-0.057
	-0.234	-0.479
	0.191	-0.503
	-0.495	-0.580
	0.303	-0.584

ANOVA						
Source of Variation	SS	df	MS	F	P-value	Fcrit
Clumping	0.016	1	0.016	0.197	0.663	4.494
Amount	0.417	1	0.417	5.292	0.035	4.494
Interaction	0.002	1	0.002	0.019	0.892	4.494
Within	1.262	16	0.079			
Total	1.696	19				

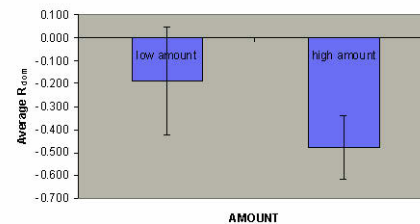


Figure 4: Analysis of Dominance Reciprocity under 4 Environmental Conditions (small population)

lish what factors and dynamics are responsible for emergent social patterns.

If patterns of behaviour happen to correspond to those observed natural systems new hypotheses for existing explanations may be derived from the model. Such hypotheses are often counter-intuitive and innovative. SimianWorld studied the impact of food availability (profuse/scarse) and distribution (clumped/dispersed) on the types of affective, dyadic relationships agents developed. This was based on the hypothesis that competition for resources in an environment influences the types of society (egalitarian or despotic) that emerge - results were statistically significant for reciprocity of social relationships when there is a large amount of food available and the population is relatively small. Negative reciprocation of dominance interactions implied that agents initiated a greater number of aggressive interactions than they received indicating a more despotic social structure. It is planned to include in the model a second order modifier in the form of a hormone-like mechanism. This will introduce a more biologically plausible sigmoid rather than a linear decrease/increase of the levels of energy and sociability, and will be designed to influence levels of aggressive versus friendly behaviour which characterise the types of social relationship of egalitarian or despotic societies. It is hoped that results will show statistical significance in tests for reci-



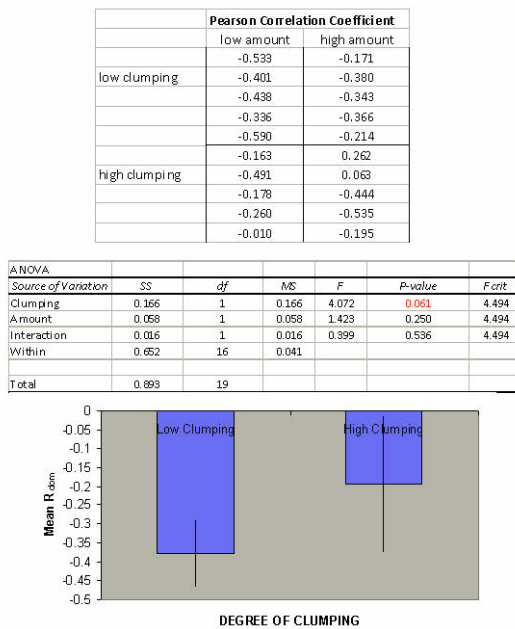


Figure 5: Analysis of Dominance Reciprocity under 4 Environmental Conditions (large population)

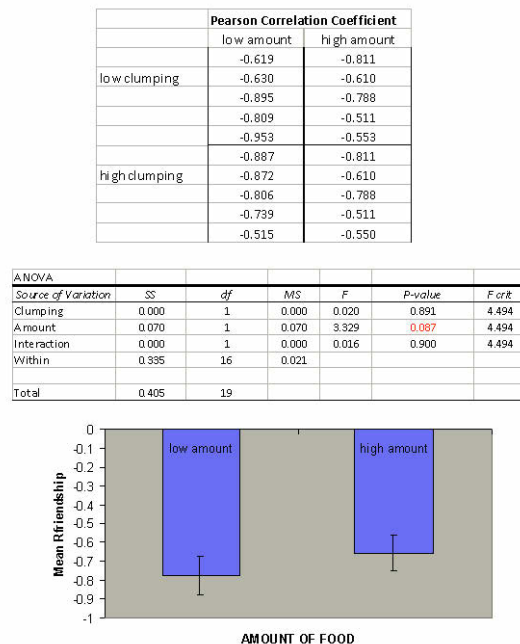


Figure 6: Analysis of Friendship Reciprocity under 4 Environmental Conditions (large population)

procuity and bi-directionality of social relationships.

## References

- Cañamero, L. (1997). Modeling motivations and emotions as a basis for intelligent behaviour. In Johnson, W., editor, *Proceedings of the First International Symposium on Autonomous Agents*, pages 148 – 155.
- Cañamero, L. (2001). Emotions and adaptation in autonomous agents: A design perspective. *Cybernetics and Systems: An International Journal*, 32 : pages 507 – 529.
- de Waal, F. B. M. and Ferrari, P. (2010). Towards a bottom-up perspective on animal and human cognition. *Trends in Cognitive Sciences*, 14 : pages 201 – 207.
- Hemelrijk, C. (1999). An individual-oriented model on the emergence of despotic and egalitarian societies. *Proceedings of the London Society*, 266 : pages 361 – 369.
- Hemelrijk, C. (2004). The use of artificial-life models for the study of social organization. In Thierry, B., Singh, M., and Kaumanns, W., editors, *Macaque Societies. A Model for the Study of Social Organization*, pages 295 – 313. Cambridge University Press.
- Hemelrijk, C. and Puga-Gonzalez, I. (2012). An individual-oriented model on the emergence of support in fights, its reciprocation and exchange. *PLOS ONE*, 7.
- Hogeweg, P. (1988). Mirror beyond mirror, puddles of life. In Langton, C., editor, *Artificial Life, SFI Studies in the Sciences of Complexity*, pages 297 – 316. Addison Wesley.
- te Boekhorst, R. and Hogeweg, P. (1994). Self-structuring in artificial “CHIMPS” offers new hypotheses for male grouping in chimpanzees. *Behaviour*, 130 : pages 229 – 252.
- Thierry, B. (2006). The macaques: A double-layered social organization. In Campbell, C., Fuentes, A., MacKinnon, K., Panger, M., and Bearder, S., editors, *Primates and Perspective*, pages 195 – 218. Oxford University Press.
- Thierry, B., Singh, M., and Kaumanns, W. (2004). *Macaque Societies. A Model for the Study of Social Organization*. Cambridge University Press.
- Tomasello, M. and Call, J. (1997). *Primate Cognition*. Oxford University Press.
- Tyrrell, T. (1993). *Computational Mechanisms for Action Selection*. Centre for Cognitive Science, University of Edinburgh.
- van Hooff, J. and van Schaik, C. (1992). Coalitions and alliances in humans and other animals. In Harcourt, A. and de Waal, F., editors, *Cooperation in Competition: the Ecology of Primate Bonds*, pages 195 – 218. Oxford University Press.
- van Schaik, C. (1983). Why are diurnal primates living in groups? behaviour. *Behaviour*, 87 : pages 120 – 144.
- van Schaik, C. (1989). The ecology of social relationships amongst female primates. In Standen, V. and Foley, R., editors, *Comparative Socioecology, the behavioural ecology of humans and other mammals*, pages 195 – 218. Oxford: Blackwell.

- Vehrencamp, S. (1983). A model for the evolution of despotic versus egalitarian societies. *Animal Behaviour*, 31 : pages 667 – 682.
- Wrangham, R. (1980). An ecological model of female-bonded primate groups. *Behaviour*, 75 : pages 262 – 300.
- Wrangham, R. (1987). Evolution of social structure. In Smuts, B., Cheney, D., Seyfarth, R., Wrangham, R., and Struhsaker, T., editors, *Primate Societies*, pages 195 – 218. University of Chicago Press.

# Self-Organisation of Generic Policies in Reinforcement Learning

Simón C. Smith and J. Michael Herrmann

Institute of Perception, Action and Behaviour, School of Informatics  
The University of Edinburgh, 10 Crichton St, Edinburgh, EH8 9AB, U.K.  
artificialsimon@ed.ac.uk, michael.herrmann@ed.ac.uk

## Abstract

We propose the use of an exploratory self-organised policy to initialise the parameters of the function approximation in the reinforcement learning policy based on the value function of the exploratory probe in a low-dimensional task. For a high-dimensional problems we exploit the property of the exploratory behaviour to establish a coordination among the degrees of freedom of a robot without any explicit knowledge of the configuration of the robot or the environment. The approach is illustrated by a learning tasks in a six-legged robot. Results show that the initialisation based on the exploratory value function improve the learning speed in the low-dimensional task and that some correlation towards a higher reward can be acquired in the high-dimensional task.

## Introduction

Reinforcement learning aims at solving dynamical optimisation problems which may be formulated in terms of discrete or continuous variables (Sutton, 1988; Sutton and Barto, 1998; Doya, 2000). It is based on an utility function and/or the construction of a control policy such that optimal performance can be reached asymptotically under certain conditions. Particularly, in continuous time and space the use of function approximation is imperative to match the complexity of the problem. Various techniques have been proposed in order to approximate the relevant functions, e.g. kernel-based methods (Xu et al., 2007; He et al., 2011), normalised Gaussian networks (Sato and Ishii, 2000; Doya, 2000), Fourier basis function (Konidaris et al., 2008) and echo state networks (Jaeger, 2001; Szita et al., 2006).

The initialisation of the parameters of the function approximator becomes non-trivial when — as in most robotic tasks — a high learning speed is required. Ideally, the initialisation should be such that the learning trajectory can follow the gradient without being trapped in undesired optima. Often, the approximator is initialised by small random values, which in some cases aids the exploration of the state and action space. Other approaches assign optimistic value to every position of the state space which also provides an initial incentive to explore until values in a more realistic range are found. Often, however, these value will not be

close to the true expected future rewards, because the optimistic values decay linearly while sufficiently exploration takes usually longer, such that, at least in the more complex problems, more flexible exploration strategies are worth being considered.

We propose a self-organising exploration mode that will discover coherent behaviour in a robot in an autonomous learning stage before reward signals are used or are available. This behaviour will be used here to pre-shape the parametric representation of the policy in an actor-critic reinforcement learning scheme. The exploration method produces an on-policy estimate of the value function, such that its value can be a good indicator of how well the robot will perform in a specific task later if only the promising actions of the policy are used. The exploration will in particular introduce a bias that can reduce the complexity of the problem by using information that was inexpensively obtained earlier. In our robotic application this means that the robot preferentially guided to regions in the state space where controllability and predictability of the dynamics is high.

Nevertheless, function approximation does not easily generalise to high dimensions unless independence or hierarchical structures can be assumed. In robotic problems as well as in biological examples, however, such assumptions are rarely justified, i.e. often the exploitable structure is not explicitly known. As the main contribution of this study, we propose a combination of homeokinetic and reinforcement learning which uses for high-dimensional reinforcement learning tasks a combination of autonomous exploration with a reward-weighted extraction of information.

In the case that the exploitable structure is known in advance, a similar effect has been shown before (Martius and Herrmann, 2011). In this study, it was shown for a track-like robot (“armband”) that the learning time can decrease even for an increase of the mechanical complexity of the robot if the complexity of the control problem was relatively low. The reason for this observation was in addition to the built-in interaction structure that the robot was less likely to self-obstruct in the high-dimensional case. The speed-up saturated at a few tens of dimensions and the remaining learning

time was low due to the homogeneity of the robot's configuration. Here we will study a more complex problem, namely a hexapod with twelve degrees of freedom which require a measure of coordination for ambulation or navigation.

The ambivalence of training and self-organisation reflects an important principle in biological learning. Although in many cases the external event distribution is sufficient to drive learning successfully, there are often intrinsic mechanisms available as a more or less equally successful fall-back option which the organism can rely on when the environment deviates from the evolutionarily anticipated standard. We will not discuss, how the organisms deal with the unfortunate latter case, but with the potential benefits of a preparation before environmental reward signals are available or while they are not yet critical such as in play in a protected environment. In addition to this consideration and particularly in robotic applications, the prior-learning scheme can add naturalness to the movements and simplify the search space when the purposeful movements are to be learned subsequently.

The early-learning algorithm relies on a self-organising control paradigm (Martius et al., 2007). This controller creates coherent exploratory behaviour by maximising the predictability of the robot action at the same time that it tries to maximise the sensitivity of each motor command. In order to propagate the best actions from the exploratory mode to reinforcement learning, we let the value function to be learned by the critic while the actor is fixed to the exploratory policy. The on-policy property of the actor-critic algorithm, i.e. the fact that the value function is learned based on the actual policy, makes this method suitable to assess the performance of the exploratory regime. In a first low-dimensional experiment the exploration policy is propagated directly to the actor's policy when the value function is positive. For negative values of the value function we propagate the opposite actions. In reinforcement learning the value function invokes the beneficial actions but it gives us little information about where to explore next if its value is not sufficient for the task. Another experiment is realised with a high-dimensional case. In order to overcome the curse of dimensionality, a closed-loop controller is learned which can function similar to a central pattern generator (CPG), where its coordination factors are shaped following the instantaneous reward.

We present a comparison of our approach with a standard version of continuous reinforcement learning (Doya, 2000) in low-dimensionality, whilst in high-dimensionality the direct reward is used to propagate the correlation between the degrees of freedom. The reward of the tasks is the horizontal speed of a six-legged robot.

## Reinforcement learning in continuous domains

For continuous reinforcement learning (Doya, 2000), we will have to adjust the weights  $w^A$  that determine the output

$u$  of a controller  $U$  which is given by

$$u_t = U_t(x_t) = s(A(x_t; w^A) + \sigma n_t), \quad (1)$$

where  $s$  is usually a sigmoidal or an identity output function,  $n$  is a probing input signal of strength  $\sigma$  and

$$A(x_t; w^A) = \frac{1}{N(x_t)} \sum_i w_i^A \exp\left(-\frac{\|x_t - \mu_i\|^2}{2\rho_i}\right) \quad (2)$$

represents the approximator function with parameter  $w^A$  of the actor's policy. The values of  $\rho_i$  and  $\mu_i$  represent the size and centre of a basis function, here are assumed to be fixed. The factor  $N(x_t) = \sum_i \exp\left(-\frac{\|x_t - \mu_i\|^2}{2\rho_i}\right)$  normalises the output. The parameters  $w^A$  are updated according to

$$\dot{w}_i^A = \varepsilon_A \delta_t n_t \frac{\partial A(x_t; w^A)}{\partial w_i^A}, \quad (3)$$

where  $\varepsilon_A$  is the actor's learning rate. The last term in Eq. 3 can be obtained directly from the explicit form of the policy in Eq. 2. The essential part of the learning rule includes the correlation of the probing input  $n$  and the delta error,

$$\delta_t = r_t - \frac{1}{\tau} V_t + \dot{V}_t, \quad (4)$$

where  $r_t$  is the instant reward at time  $t$ ,  $\tau$  is the time constant for discounting future rewards and the utility function  $V$  is approximated by another parametrised function which is updated based on the approximation of the critic by the relation

$$\dot{V}_t \cong (V_t - V_{t-\Delta t})/\Delta t$$

which can be obtained from Eq. 4. The update of the parameter  $w_i^V$  of  $V$  follows the gradient descent with respect to  $\delta$ ,

$$\dot{w}_i^V = \varepsilon_V \delta(t) \frac{\partial V(x_{t-\Delta t}; w^V)}{\partial w_i^V}, \quad (5)$$

with  $\varepsilon_V$  learning rate.

The alternatives for choosing the probing signal of the robot control in Eq. 1 range from the use of noise (Gullapalli, 1990) to high-frequency oscillatory modulations of the motor command (Wiener, 1948). Our experiments (Smith and Herrmann, 2012) confirm that the type of the probe does not matter in low-dimensional problems. The dynamics of the correlation among the degrees of freedom of the controlled system becomes crucial for robots with many degrees of freedom, such that the choice of the probing stimulus becomes non-trivial. In high-dimensional problems it is not possible to test all actions in all states infinitely often as it would be required in discrete reinforcement learning algorithms. Also for continuous algorithms orienting the exploration to promising directions is essential. We propose to use an approach in the present context that has previously developed in a different setting (Martius, 2010).



## Learning in motor space

As exploration signal we propose the exploratory controller

$$\mathbf{y}_t = K(\mathbf{x}_t) = g(C\mathbf{x}_t + \mathbf{c}). \quad (6)$$

This controller receives the current sensory input vector  $\mathbf{x}_t \in \mathbb{R}^n$  and determines the direction of exploration in dependence on the multidimensional parameters  $C \in \mathbb{R}^{m \times n}$  and  $\mathbf{c} \in \mathbb{R}^m$  and the nonlinear function  $g$ , where  $\mathbf{y}_t \in \mathbb{R}^m$ . In order to adapt the parameters  $C$  and  $\mathbf{c}$ , the new sensory inputs are compared with a prediction  $\hat{\mathbf{x}}_t \in \mathbb{R}^n$  by a world model  $M$  based on previous inputs or outputs. For simplicity, we use a linear predictor that uses only the motor commands from Eq. 6 and receives thus information about previous inputs only indirectly,

$$\hat{\mathbf{x}}_{t+1} = M(\mathbf{y}_t) = D\mathbf{y}_t + \mathbf{d}, \quad (7)$$

where  $D \in \mathbb{R}^{n \times m}$  and  $\mathbf{d} \in \mathbb{R}^n$ .

The comparison of the corresponding sensory input  $\mathbf{x}_{t+1}$  and its estimate by the internal model  $\hat{\mathbf{x}}_{t+1}$  results in the prediction error  $\boldsymbol{\xi}_{t+1} = \hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}$  which is a vector in the perceptual space where  $\boldsymbol{\xi}_t \in \mathbb{R}^n$ .

In order to formulate a learning rule for the exploratory controller of Eq. 6, we will follow the procedure in (Martius, 2010) and express the error in the motor space which can be achieved by defining a transformed error  $\boldsymbol{\eta}_t \in \mathbb{R}^m$  via

$$M(\mathbf{y}_t) + \boldsymbol{\xi}_{t+1} = M(\mathbf{y}_t + \boldsymbol{\eta}_t). \quad (8)$$

Because  $M(\mathbf{y}_t) + \boldsymbol{\xi}_{t+1} = \mathbf{x}_{t+1}$ , the motor error  $\boldsymbol{\eta}_t$  can be interpreted as the control correction required to compensate the inaccuracy of the model  $M$ . The vector  $\boldsymbol{\eta}_t$  is a retrospective error that can be determined only after the event of receiving the new stimulus  $\mathbf{x}_{t+1}$ . Nevertheless, minimisation of  $\boldsymbol{\eta}$  is a relevant goal for the adaptation of the system. The definition in Eq. 8 is implicit and may be empty which calls for the use of a regularised inverse of  $M$  to explicitly obtain an approximation of  $\boldsymbol{\eta}$ . Practically, Eq. 8 is transformed into a motor level error exploiting the assumed linearity of the model in Eq. 7,

$$\boldsymbol{\eta}_t = M'^+ \boldsymbol{\xi}_{t+1}, \quad (9)$$

where  $M'^+$  is the pseudo-inverse of the derivative of the model in Eq. 7, i.e. the pseudoinverse of  $D$ . In analogy to (Der et al., 2002) this defines a homeokinetic error function in the motor space

$$E_t = \boldsymbol{\eta}_t^\top (J_t J_t^\top)^{-1} \boldsymbol{\eta}_t \quad (10)$$

where  $J$  is the Jacobian of the sensorimotor loop, see below. We are going to perform a gradient descent with respect to this error function in order to adapt the parameters of the controller defined in Eq. 6.

To calculate the Jacobian, we use the derivatives  $M'_y = D$  and  $K'_x = g' \circ C$ , with  $\circ$  defined as element-wise multiplication, such that we find from  $J_t = g'_t \circ C_t D_t = g'_t \circ R_t$ , with  $R_t \in \mathbb{R}^{m \times n}$  and  $R_t = C_t D_t$ . This gives rise to the following formulation of the shift  $\boldsymbol{\nu}$ , i.e. the change in motor command that would have been required to correctly predict the following motor command, namely

$$\boldsymbol{\nu}_{t-1} = J_t^{-1} \boldsymbol{\eta}_t.$$

While the interpretation of  $\boldsymbol{\eta}$  (Eq. 9) as retrospective error connects sensor and motor space, we have here a connection between the two points in time within the motor space that reflects the dynamical properties of the full sensorimotor loop. The error function in Eq. 10 becomes thus simply

$$E_t = \boldsymbol{\nu}_{t-1}^\top \boldsymbol{\nu}_{t-1}$$

which lead to a convenient update rule of the controller matrix  $C$ . Omitting the time indices we find

$$\begin{aligned} \frac{1}{\varepsilon_C} \Delta C &= -\frac{\partial E}{\partial C} = -2\boldsymbol{\nu}^\top \frac{\partial \boldsymbol{\nu}}{\partial C} \\ &= 2\boldsymbol{\nu}^\top J^{-1} \frac{\partial J}{\partial C} J^{-1} \boldsymbol{\eta} - 2\boldsymbol{\nu}^\top J^{-1} \frac{\partial \boldsymbol{\eta}}{\partial C} \end{aligned}$$

using the rule  $\frac{\partial Y^{-1}}{\partial X} = -Y^{-1} \frac{\partial Y}{\partial X} Y^{-1}$ . The derivative  $\frac{\partial \boldsymbol{\eta}}{\partial C}$  cannot be determined, because we have no information of the dependence of the prediction error on the controller parameters, therefore we set  $\frac{\partial \boldsymbol{\eta}}{\partial C} = 0$  and are left with

$$\frac{1}{\varepsilon_C} \Delta C = 2\boldsymbol{\nu}^\top J^{-1} \frac{\partial J}{\partial C} J^{-1} \boldsymbol{\eta} = 2\boldsymbol{\nu}^\top J^{-1} \frac{\partial J}{\partial C} \boldsymbol{\nu}$$

where

$$\frac{\partial J_t}{\partial C} = \frac{\partial}{\partial C} \left( \frac{\partial D}{\partial \mathbf{x}} + g'_t \circ C_t \right) D_t.$$

We may ignore the effect of the controller on the sensitivity of the actor in the reinforcement learning component, i.e. set  $\frac{\partial}{\partial C} \frac{\partial D}{\partial \mathbf{x}} = 0$ . We may also assume that the details of the actor are not specified by the reward but will follow essentially the homeokinetic control. In this case the term  $\frac{\partial}{\partial C} \frac{\partial D}{\partial \mathbf{x}}$  is parallel to the remainder and the resulting numerical factor can be absorbed into the learning rate. We have thus arrived at essentially the same learning rule as in (Martius, 2010),

$$\frac{1}{\varepsilon_C} \Delta C = \boldsymbol{\chi} (D\boldsymbol{\nu})^\top - \boldsymbol{\chi}^\top \frac{\partial g'^{-1} \circ \boldsymbol{\eta}}{\partial C},$$

where  $\varepsilon_C$  is a learning rate and  $\boldsymbol{\chi} \in \mathbb{R}^m$  as  $\boldsymbol{\chi} = R^{-1\top} \boldsymbol{\nu}$ .

Inserting the correct time indexes we obtain

$$\begin{aligned} \frac{1}{\varepsilon_C} \Delta C_t &= \boldsymbol{\chi}_{t-1} (D_t \boldsymbol{\nu}_{t-2})^\top \\ &\quad - 2(\boldsymbol{\chi}_{t-1} \circ g_{t-2} \circ (g'_{t-2})^{-1} \circ \boldsymbol{\eta}_{t-1}) \boldsymbol{\chi}_{t-2}^\top, \quad (11) \end{aligned}$$

with  $\chi_{t-1} = (R_t^\top)^{-1} \nu_{t-2}$ . The update rule for  $c$  can be found similarly,

$$\frac{1}{\varepsilon_C} \Delta c_t = -2 \left( \chi_{t-1} \circ g_{t-2} \circ (g'_{t-2})^{-1} \circ \eta_{t-1} \right). \quad (12)$$

### Direct learning of the actor

The exploratory controller presented above can provide a variety of coherent behaviours based solely on the interaction of the agent and its environment. We propose that such behaviours can be used to shape the action space for an actor-critic reinforcement learning problem by shaping the structure of the search space.

Initially, the agent is controlled by the homeokinetic controller (Eqs. 6, 11 and 12) giving the motor signal to the agent and also shaping the action space. Following homeokinetic motor command in Eq. 6, the implementation of the actor in reinforcement learning shown in Eq. 2 with learning rule from Eq. 3, the difference between the actual motor command and the actor's output  $e_t^A \in \mathbb{R}^m$ ,  $e_t^A = \mathbf{y}_t - A(\mathbf{x}_t; \mathbf{w}^A)$ , gives rise to the objective function,  $E_t = \frac{1}{2} \|e_t^A\|^2$ . The weights  $\mathbf{w}^A$  of the actor's approximator are updated with a gradient descent algorithm,  $\dot{w}_i^A = -\varepsilon_H \frac{\partial E}{\partial w_i}$  with  $\varepsilon_H$  the learning rate.

After the adaptation, the function approximator has stabilised, the reinforcement learning algorithm is activated and the policy calculated from the actor following Eq. 1, and the actor's parameters are updated based on Eq. 3 using random noise as probing signal.

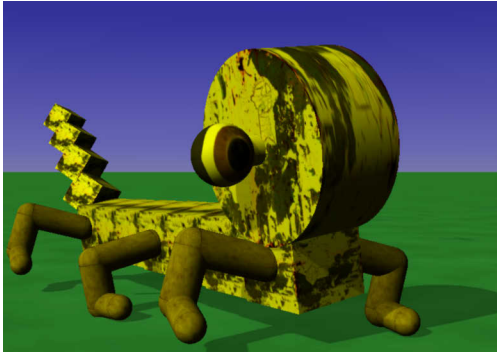


Figure 1: Simulated robot in the lpzrobots simulation environment. The design of the robot is inspired by M. C. Escher's lithograph *Wentelteefje* (1951).

### Self-organisation for parameters initialisation

In order to test the described approach, we will study a low-dimensional control problem for a simulated six-legged robot, see Fig. 1. Switching from the initialisation mode to reinforcement learning is triggered by the amplitude of the error in the approximation which is required to be below a threshold for a certain time. The propagated values from the self-organised policy to the initialisation of the actor's policy

is directly translated when the value function has a positive value as this part of the policy already shows the suitability to perform the task. When the value function is negative we propagate the opposite values of the self-organised policy. The justification for this is that the reward only tells what is a beneficial (we want to propagate) and what is not (we only know that this behaviour should not be propagated), since we have all the action state to choose from (except from the actual not beneficial behaviour) we assume that the opposite of the actual command is a better guess than a random action. This will carry the coherence found by the probing but in the opposite direction in our servo motor robot. To illustrate this point we present a toy example where the reward is directly related to the y-axis position of one leg of the robot. In Fig. 2a, the random initialisation of the reinforcement learning policy can be seen, in Fig. 2b, the shape of the exploration signal, in the Fig. 2c, the value function of derived from the exploration policy and in Fig. 2d, the propagated values from the exploration signal to the initial conditions of the actor's policy. The values of the exploration signal that have positive (blue) value function are propagated directly while the exploration signal with negative (red) value function is inverse propagated.

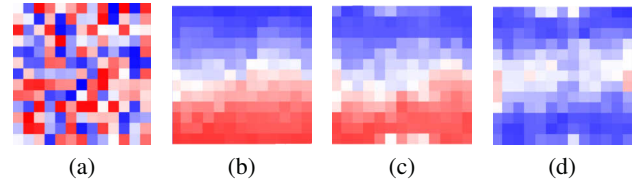


Figure 2: Shape of the approximation of the actor. The x-axis of the position of the leg is represented in the horizontal axis, the y-axis is represented in the vertical axis. For (a), (b) and (d) the colour represents a motor command with blue values closer to 1 and red values closer to -1; for (c) the colour represents the value function with same range. Figure (a) represents the random initialisation, (b) is learned from the homeokinetic controller, (c) is the value function and (d) the initialised actor's policy based on the exploratory signal following the value function. It can be seen how the propagated values from (b) to (d) depend on (c).

In the low-dimensional set-up the task of the robot will be to walk forwards as fast as possible and the rewards will be directly proportional to the absolute value of the velocity of the centre of mass of the robot. A virtual leg is trained and will form a CPG whose motor signal is transmitted to the rest of the limbs either as an in-phase or as an anti-phase signal. While the random initialisation in of the two degrees of freedom may lead to local minima or slow convergence, a smoother function is brought about due to the training with the homeokinetic controller. This may allow for a faster learning once the information about the task is available by the reward signal.

The results shown in Fig. 3 demonstrate that the homeokinetic learning indeed improves the performance in the learning task.

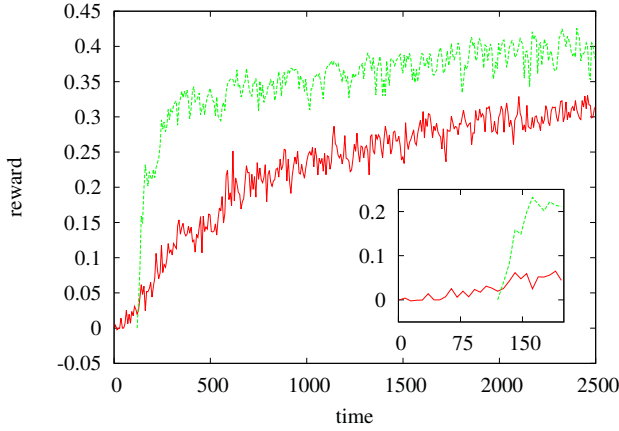


Figure 3: Results of reinforcement learning with random initialisation (red continuous line) of the parameters and with parameters shaped by homeokinetic controller (green dashed line). The six-legged robot receives reward based on horizontal speed. For the green dashed curve the first 600 seconds (shown as the first 120 point of averaged speed over 5 seconds) are used to pre-train the robot, i.e. no reward is available, which can be seen in detail in the inset. With the homeokinetic pre-training, the time to achieve the highest velocity achievable by reinforcement learning is significantly decreased. If the two systems continue to learn for much longer times, our experiments show that both arrive at a very similar reward level.

The basis for the comparison is the reinforcement learning initialised with random weights in the function approximator. An increment in the learning speed can be noticed as a result of the exploratory learning due to homeokinetic control.

### Reward-weighted correlation

A more flexible method will be discussed in the following as a generalisation of the previous approach. At the same time we generalise the variants in (Martius and Herrmann, 2012) by including the reward signal in the extraction of the interaction structure. We consider the correlation between sensory inputs and motor commands,

$$W_{ij} = \frac{\langle (x_{i,t} - \langle x_i \rangle)(y_{j,t} - \langle y_j \rangle) \rangle}{\sqrt{\langle (x_{i,t} - \langle x_i \rangle)^2 \rangle \langle (y_{j,t} - \langle y_j \rangle)^2 \rangle}}, \quad (13)$$

where  $\langle \cdot \rangle$  denotes a sliding temporal average with time constant  $\tau_W$ . Eq. 14 can be transformed into a reward-related quantity by an appropriate weighting based on the rectified

reward signal  $r^{[+]}$ .

$$W_{ij,t}^{r^{[+]}} = \frac{\langle r_t^{[+]}(x_{i,t} - \langle x_i \rangle)(y_{j,t} - \langle y_j \rangle) \rangle}{\sqrt{\langle (r_t^{[+]} - \langle r^{[+]} \rangle)^2 \rangle \langle (x_{i,t} - \langle x_i \rangle)^2 \rangle \langle (y_{j,t} - \langle y_j \rangle)^2 \rangle}}, \quad (14)$$

As before the reward signal  $r$  is determined by the forward speed of the centre of mass of the robot. The factor  $r^{[+]}$  equals  $r$  for positive forward speed and is zero if the robot is actually moving backwards. In this way only those sensori-motor couplings that directly contribute to the reward enter the average. The control weights are a smoothed version of the result of Eq. 15.

$$\bar{W}_{ij,t+1} = \varepsilon_W W_{ij,t}^{r^{[+]}} + (1 - \varepsilon_W) \bar{W}_{ij,t}. \quad (15)$$

where  $\varepsilon_W < 1$  is the adaptation rate.

### Learning gait patterns in a hexapod

In the high-dimensional task, instead of learning with a classic reinforcement learning approach we try to discover the correlation between the different degrees of freedom based on the instantaneous reward. The exploration is produced by the homeokinetic controller and the learning rule is based on Eq. 16. In the following experiment we compare the results obtained from the behaviour of the robot for differently obtained controller matrices in a close-loop setting. Closed-loop feedback control is realised by a controller output related to the current input via  $y_t = Hx_t$ , where  $x_t$  and  $y_t$  are the input sensors and output motor commands vectors respectively. In the first case served as a baseline, the matrix  $H$  is obtained from a hand-crafted CPG matrix that was designed to control the robot in a smooth and highly rewarded fashion. The CPG matrix can be used to perform an open-loop control of the robot, but by a minor phase shift it functions also in closed-loop. In the second case the feedback matrix was learned from the correlations observed in the first case (Eq. 14), i.e. without taking the reward in consideration, while in the third case the matrix was learned by the robot while exploring based on a homeokinetic controller. The three matrices are shown in Fig. 4, where the sensor inputs are presented as rows and the motor commands as columns. All matrices are scaled appropriately such that resting state becomes unstable and the legs of the robot start to move.

In order to characterise the behaviour generated in these cases we show in Fig. 5 the behaviour of one leg of the robot with its  $x$  and  $y$  position. Fig. 6 is added to show the phase relations between the two degrees of freedom for one leg. The CPG-style matrix produces the desired behaviour for the legs with a tripod gait and maximising the use of the leg state space. The second matrix inherits from the behaviour of the first case a consistent relation between the degrees of freedom, however, the matrix is blurred due to hardware-induced deviation from the ideal interaction matrix. In some cases white boxes representing the absence of a correlation,

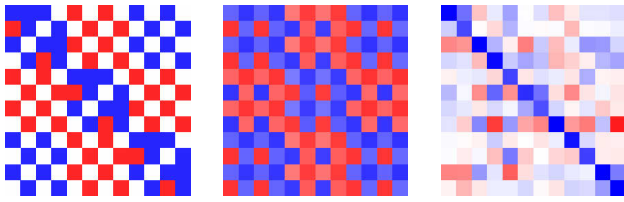


Figure 4: The matrix on the left contains the coefficients similar to a coupled CPG, which is sufficient to perform a tripod gait. With the small changes this matrix can also be used as a closed-loop controller. The matrix on the middle has been learned by the system actuating over a close-loop disregarding the reward. On the right is the re-estimated matrix that was obtained from Eq. 15 while the robot was exploring with the self-organised probing signal. In Fig. 9, the reward obtained by the left and middle matrices can be seen.

while in other cases the correlations values appear blurred, see the centre image in Fig. 4.

The third matrix is learned based on the reward (Eq. 16) while exploring using the homeokinetic adaptation rule. Correlations in one leg can be seen in the bottom graphic of Fig. 5, this behaviour has been learned within a small time of exploration and the rotational displacement of the leg can be seen in the blue line in Fig. 6. The contact in the ground is less and the exploited state space is smaller, although this still generates a behaviour that is positive towards reward.

The relationship between degrees of freedom from different legs is illustrated by Fig. 7, where we show the relation between a front leg with the lateral middle leg in the horizontal direction. The tripod gait generated by the CPG -the upper and middle graph of the figure- shows the expected phase relations. The same relation is observable in the learned matrix outlining an incipient tripod gait. This generation of the later behaviour is not influenced by the designed matrix which is shown here only for comparison.

The result of the longer experiment can be seen in Fig. 8 where the averaged reward of the tripod gait produced by a designed matrix, the homeokinetic controller, and the learned from reward matrix has been collected for 30 minutes of closed-loop exploitation. The reward of the CPG-style matrix is consistent and positive for all the experiment as expected. The homeokinetic reward is small and also consistent in time, since this controller is not promoted to follow any specific action other than explore coherently. The bigger amplitude of the reward in the learned approach is interpreted as the robot behaving with a bigger variety of actions that tends to maximise the reward but still not completely removing all the actions that leads to a negative reward behaviour. The relations shown in Figs. 5, 6 and 7 holds for some of the degrees of freedom but not for all of them. The final behaviour of the robot produces movement to the front and to the side as well. It can be seen that the homeokinetic

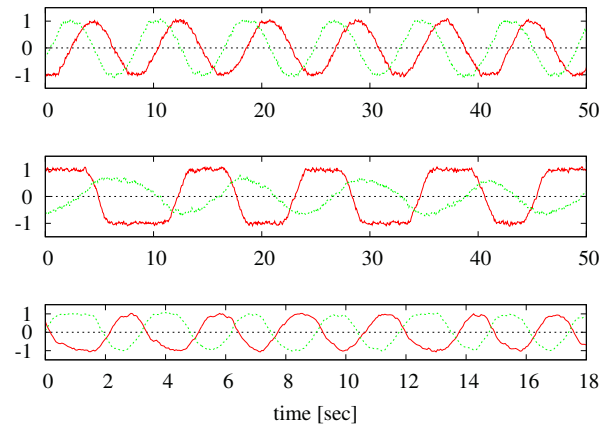


Figure 5: Considering a single leg controlled by a designed coupling matrix (top) we observe a phase shift between the horizontal (red continuous line) and vertical (green dashed line) actuation pattern. The movement of the leg (middle) does not follow the trajectory precisely, but keeps a similar phase shift. Using the present approach (bottom, Eq. 16) the movement pattern becomes more smooth which may point to a reduced energy consumption, but the phase shift has increased, the speed of the robot (as implied by the guidance matrix  $W^{r[+]}$  Eq. 15) being in the same range, see Fig. 8

exploration produces a small quantity of reward so the captured behaviours are an average of good but still not maximum rewarded actions. Note that the learned matrices have been normalised and multiplied by a factor in order to make the robot responsive in the closed-loop mode, this is required as the acquired results tend to be not big enough given the averaging nature of the approach.

## Discussion

We should note that the effect of the discovered structure may not always be beneficial for the robot by itself, but the potential misguidance can be diminished by a manipulation of the value function. As the shape of the robot's body distinguishes one of the directions of movement, there is also a bias in the exploration towards the forward direction. If the goal was instead to move backwards, then our algorithm would fail to provide a direct advantage, but the propagation of the opposite policy values of the policy may still provide a better starting point than random initialisation. Nevertheless, our results confirm that even if the exploration does not directly bring about a coherent behaviour that will receive high reward, it can still induce an acceleration of the learning of the task.

Obviously, also the learning scheme based on Eq. 15 will not be effective for all systems and that more complex relations between reward and sensorimotor coupling than studied here are clearly possible, but it is not the goal to im-



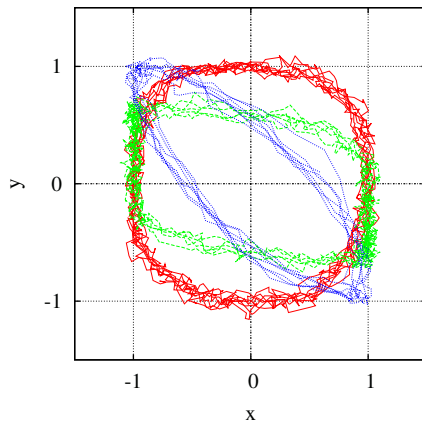


Figure 6: Configuration space representation of the trajectories from Fig. 5. The red continuous line represents the CPG-style matrix, the green dashed line is for the behaviour learned by following the first case without reward and the blue-dotted line represents the rotation learned by the system following Eq. 16.

pose these relations precisely, but rather to introduce a bias into the self-organising system such that any deviations between the true sensorimotor couplings and the relation that is implied by the guidance matrix  $W_{r^{[+]}}$  are resolved by the exploratory behaviour of the self-organising controller. We should remark, however, that a substantial deviation between guidance and realisable behavioural modes may compromise the efficiency although usually not the effectiveness of the control.

The use of the rectified reward signal in Eq. 15 avoids a critical step in the low-dimensional case that was considered in the first part. If the reward signal is negative, taking the opposite action might not always be beneficial or even possible. We have, thus implicitly assumed in the first part, that the opposite action is meaningful and in the second part that all positive rewards are actually relevant for the task which is not required in many other algorithms where only difference of reward signals enter.

## Conclusions

We have studied an exploratory self-organised mechanism for discovering promising initialisations for a parametrised policy and to establish coordination among the controllable degrees of freedom of a robot. We used a homeokinetic controller that is based on sensible and predictable exploration and does not require explicit knowledge of the robot's configuration or the environment. The approach is illustrated by a low and a high dimensional tasks implemented in a six-legged robot. The results imply that the initialisation of the parameters for the function approximation by a self-organised approach improves the learning of the proposed

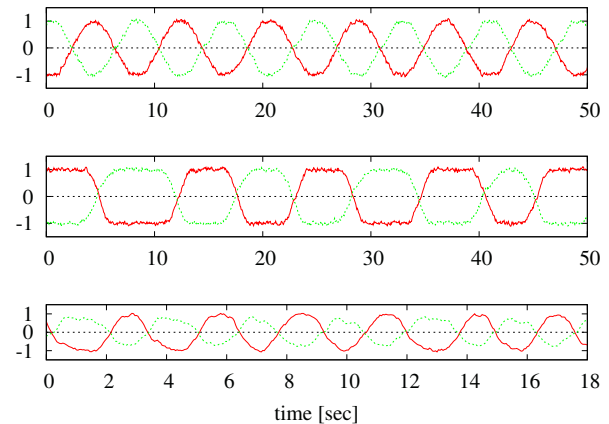


Figure 7: In tripod gait, opposite legs follow an antiphase movement (red continuous line represents front-right leg and green dashed line represents middle-left leg horizontal position), which can be enforced by a designed matrix (top). A similar pattern is discovered by the system following the designed matrix (middle). For the present approach (bottom, Eq. 16) the movement pattern is similar but does not reproduce the same trajectory for all expected DoF as in the designed matrix.

tasks and that in high-dimensional set-up the correlation between degree of freedom can be acquired to improve the long-term reward.

## Acknowledgements

S.C.S. has been funded by *The Advanced Human Capital Program of the National Commission for Scientific and Technological Research (CONICYT) of the Republic of Chile*. J.M.H. was partially funded by the BMBF in the framework of the *Bernstein Focus for Neurotechnology Göttingen*, grant number 01GQ0811.

## References

- Der, R., Herrmann, J. M., and Liebscher, R. (2002). Homeokinetic approach to autonomous learning in mobile robots. *VDI-Berichte*, 1679:301–306.
- Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural Computation*, 12:219–245.
- Gullapalli, V. (1990). A stochastic reinforcement learning algorithm for learning real-valued functions. *Neural Networks*, 3:671–692.
- He, H.-G., Hu, D., and Xu, X. (2011). Efficient reinforcement learning using recursive least-squares methods. *arXiv preprint arXiv:1106.0707*.
- Jaeger, H. (2001). The “echo state” approach to analysing and training recurrent neural networks-with an erratum

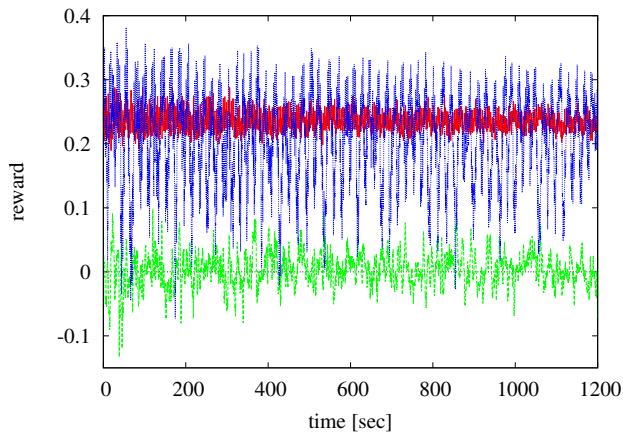


Figure 8: Results for the high dimensional case for a test run of 20 minutes. The reward is derived from the forward velocity of the hexapod, fluctuations correspond to steps. The red continuous line is the tripod gait produced by the designed matrix, the green dashed line is for behaviour obtained within the exploration mode, and the blue dotted line gives the results for a learned matrix that was sampled during homeokinetic exploration for 20 minutes.

note. Bonn, Germany: German National Research Center for Information Technology, GMD Technical Report, 148.

Konidaris, G., Osentoski, S., and Thomas, P. S. (2008). Value function approximation in reinforcement learning using the fourier basis. *Computer Science Department Faculty Publication Series*, page 101.

Martius, G. (2010). *Goal-oriented control of self-organizing behavior in autonomous robots*. PhD thesis, Göttingen University.

Martius, G. and Herrmann, J. M. (2011). Tipping the scales: Guidance and intrinsically motivated behavior. In *Proc. of Europ. Conf. on Artificial Life*, pages 766–775.

Martius, G. and Herrmann, J. M. (2012). Variants of guided self-organization for robot control. *Theory in Biosciences*, pages 1–9.

Martius, G., Herrmann, J. M., and Der, R. (2007). Guided self-organisation for autonomous robot development. In *Europ. Conf. on Artificial Life*, pages 766–775.

Sato, M.-A. and Ishii, S. (2000). On-line EM algorithm for the normalized Gaussian network. *Neural Computation*, 12(2):407–432.

Smith, S. C. and Herrmann, J. M. (2012). Homeokinetic reinforcement learning. In *Partially Supervised Learning*, pages 82–91. Springer.

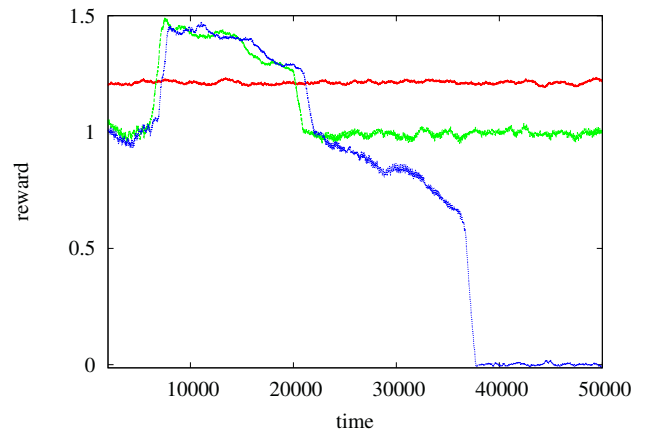


Figure 9: Performance comparison of the reward obtained by the designed matrix (left image in Fig. 4) in an open-loop setting (red continuous line) and the learned correlation matrix (middle image in Fig. 4). The green dashed line and the blue dotted one learn from the first matrix for about 1000 steps, after near 10000 steps they are combined with the designed matrix to produce the motor commands  $\mathbf{y}_t = (1 - \gamma)P\mathbf{x}_t + \gamma H\mathbf{x}_t$  where  $P$  and  $H$  are the designed and learned matrices respectively, and  $0 \leq \gamma \leq 1$  is the factor that allows different combination of the two terms. Initially, small values of  $\gamma$  are proven which favour reward. Then  $\gamma$  is incrementally increased until step 20000. A decay on reward can be seen. After this point the green dashed line continues in a closed loop ( $\gamma = 1$ ) without learning, and the blue dotted line continues in a closed loop but learning. After some steps the error accumulate and the robot stops walking.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA. A Bradford Book.

Szita, I., Gyenes, V., and Lőrincz, A. (2006). Reinforcement learning with echo state networks. In *Artificial Neural Networks (Proc. ICANN 2006)*, pages 830–839. Springer.

Wiener, N. (1948). *Cybernetics or Control and Communication in the Animal and the Machine*. Hermann Editions, Paris.

Xu, X., Hu, D., and Lu, X. (2007). Kernel-based least squares policy iteration for reinforcement learning. *IEEE Transactions on Neural Networks*, 18(4):973–992.

# ASAP: an Ant resource Search Algorithm for swarm-like P2P networks

António Homem Ferreira<sup>1</sup>, Carlos Martinho<sup>1</sup>

<sup>1</sup>INESC-ID / Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal  
{antonio.h.ferreira,carlos.martinho}@ist.utl.pt

## Abstract

The Ant Colony Optimization (ACO) meta-heuristic is a proven approach for solving complex distributed problems, being the routing problem one of such. By exploring the surroundings and indirect communication through pheromones, ants can find and follow the shortest path between a food source and its nest. Based on these characteristics, we present an ant-based algorithm for performing in-network resource search in a swarm-like Peer-to-Peer network. By marking connections that share same interests with a synthetic pheromone, a node can easily find a resource without having a significant impact on the network performance. Our approach focuses on decreasing the number of messages generated by each search, without having a negative impact on user experience. To achieve this, we present an algorithm that dynamically adapts based on the information a node has of its surroundings. The more information a node has of its neighbors, the higher the probability of choosing an exploitation strategy over an exploration one. Furthermore, the higher the number of nodes visited by an ant (and thus different paths followed), the lower the number of nodes explored in an exploration strategy. In order to decrease the number of messages sent to nodes that have already processed it, the parent ant informs each of its cloned ants about all the nodes to which each of these cloned ants will be sent to. Through simulation, we show the impact of these design choices in the algorithm's performance and discuss how it can be configured in order to adapt it to different networks.

## Introduction

Peer-to-Peer (P2P) protocols rely on decentralized architectures for providing a large number of services like VoIP, video streaming, file sharing, etc. This architecture allows P2P networks to be extremely scalable, since every node can connect and exchange resources with every other node (being storage, processing power, content, etc). Due to these characteristics, P2P protocols account for a high percentage of Internet traffic (Cisco, 2011)(Sandvine, 2011). One of the most popular architecture in P2P networks are swarms, where the resource sharing is done by grouping peers sharing a same resource in a swarm. These swarms are isolated from each other, and for a peer to share different resources, it has to participate in several different swarms. Due to this

swarm-like architecture, in-network search becomes difficult to implement. Although it is still possible to search for the resource identifier, in most cases, a peer needs to hold that resource in order to generate the identifier or search for it outside the network. This means that this identifier has to be either known at start or be generated with specific resource information that might not be known while performing a search. As for keywords search, these networks usually rely on outside services. However, there are several techniques that can be used to implement in-network search through keywords. Many of these techniques are nature-inspired algorithms that try to reproduce behaviors observed in nature. This work presents ASAP: an Ant Resource Search Algorithm for swarm-like P2P networks based on the Ant Colony Optimization (ACO) meta-heuristic (Dorigo and Caro, 1999)(Dorigo and Gambardella, 1997). By sending messages (ants) over the network and marking with a synthetic pheromone the links between nodes that lead us to a search response, we can forward future searches through those same links. By carefully choosing to whom the search messages should be forwarded to, we minimize the impact on network performance since flooding is avoided and less data is exchanged, resulting in lower bandwidth consumption in order to perform a search. As for user experience, it can also be improved since more and faster results can be achieved. Our approach focuses on dynamically adapting to the query results obtained. The algorithm was designed to evolve gradually from an exploration strategy, where a search is forwarded to many nodes in order to gather network information, to an exploitation strategy, where only the ones with the highest pheromone value receive and process the search message. However, the strategy choice also depends on the number of nodes already visited by the ant. If an ant knows that it or some clone already visited a high number of nodes, the possible number of paths the search is following increases and thus we can focus on forwarding the ant to the nodes that have a higher pheromone value. This way, we avoid flooding the network until the time-to-live (TTL) of the ant is reached. Furthermore, we also focus on minimizing the total number of messages sent between two

nodes that have already processed that same search. Discarding these messages has a significant impact on the network performance but none on the user experience.

Through simulation, we evaluate the algorithm's performance and show how the parameter configuration should be done based on some network properties. The simulated environment ran on peersim (Montresor and Jelasity, 2009) in a simulated swarm-like content sharing P2P network with a total number of 10000 nodes distributed over 50 different swarms, with each of the swarms sharing different content. As in real networks, nodes enter and leave the network throughout the experiment. Our results show the impact of the algorithm in both network performance and user experience. We show how the parameter configuration and variation affects the algorithm's performance and also show the importance and impact of some choices made during the development of the algorithm.

This paper is organized as follows: Section II gives a brief explanation on swarm-like P2P networks and Section III describes an overview of some of the related work done in this area. Then Section IV presents the algorithm, design and choices made during development, and Section V describes the simulation environment and obtained results. Finally, Section VI presents the final conclusions.

### Swarm-like P2P networks

Swarm-like P2P networks aggregate nodes that share the same resource in a swarm. These swarms are isolated and independent from each other, where the nodes that belong to it only share a single resource among themselves. However, nodes can participate in as many different swarms as they please, as long as they are willing to share different resources. These multiswarm nodes can operate as a bridge between different swarms, connecting the isolated swarms with each other. In these P2P networks, despite being aggregated into swarms, nodes are not necessarily connected to all others that participate in the same swarm. Figure 1 illustrates a swarm-like P2P network, where the lines represent the connection between two nodes.

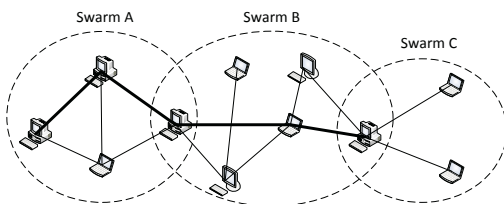


Figure 1: Swarm-like P2P network architecture.

Although proving to be a successful architecture for resource sharing, these networks usually do not provide an in-network keyword search mechanism due to their characteristics. In a swarm-like P2P network, in order for a node

to make a specific in-network keyword search, it needs to send a message with the desired keyword to all or some of its neighbors, who will then forward it until a result is found or the nodes stop forwarding the message. One of the biggest problems with this approach is its impact on network performance as it can consume a great amount of the available bandwidth if the number of messages generated by each search is too high. On the other hand, if the number of messages is too low the search algorithm might have difficulty in finding resources that the node's closest neighbors don't have. In order for the algorithm to achieve best results, it needs to have as little impact on the network as possible and, at the same time still be able to find resources in the network. Furthermore, the search algorithm also needs to be able to adapt as nodes enter and leave the network. For this reason, in-network search is usually made through the resource identifier that, in most cases, is generated based on specific resource information that a node might not have if it does not hold that resource. As for keyword search, it is usually provided by services outside the network.

Figure 1 shows the message flow for a keyword search initiated by Node 1 in swarm A. The message is forwarded from swarm to swarm by the bridge nodes (the ones that participate in multiple swarms) until it reaches a node that has that resource, then a response message is sent back through the same path to Node 1.

### Related work

The Ant Colony Optimization (ACO) meta-heuristic has been adopted by many to address complex distributed problems like routing problems in computer networks. One such work is SemAnt (Michlmayr, 2006), an ant-based algorithm to provide a distributed search engine in an unstructured P2P content sharing network. In SemAnt, each peer holds a table with a pheromone value associated with a specific keyword and a node. These keywords are predefined for the whole network. When an ant finds a desired content, it generates a backward ant that will follow the same path backward and update each corresponding pheromone value. From time to time, there is an evaporation rule that decreases the pheromone values in the tables. Though simulation, the author shows how the SemAnt outperforms the k-random walker approach (Lv et al., 2002). However, the simulated environment was based on a static network. One other work that focused on using the ACO for search in unstructured P2P networks is ERAntBudget (Wu et al., 2008). This work combined the Budget mechanism (Gkantsidis et al., 2005) and the ACO principle in an attempt to avoid generating a large amount of network traffic when searching for popular content, since there is a high probability of a query returning too many results. One other objective was to improve the query hit ratio for unpopular objects. Though simulation they compare ERAntBudget with other used techniques and showed that it can achieve a higher success rate using



less bandwidth. Later, the same authors presented a modified version called AEAntBudget (Chen et al., 2010). This new version inherited the characteristics of ERAntBudget and also provided mechanisms for progressively expanding the search scope based on previous results. With this modification, AEAntBudget can achieve better results than ERAntBudget. AntSearch (Wu et al., 2006) also adopts an ACO technique for query flooding in a P2P network. However, the authors' focus is in identifying free-riders, peers that consume but do not share their resources with the network. To do so, AntSearch uses pheromone values to identify these free-riders and avoid sending messages to the same, generating a smaller number of messages and getting query hit from peers that have a higher probability of sharing their resources with others.

## Model

### ANT colony's foraging behavior

Every living organism shows a foraging behavior which can be defined as: (1) search for a food source, (2) catch the food source and (3) ingest food source. Although it can be defined by the previous three steps, every organism has adopted and developed its own way of performing each action, optimizing it as needed. A well-known example is the ant colony. Ants by themselves are animals that do not appear to have much of an intelligent behavior, however, when in colony, they show a remarkable intelligent behavior. Ants always follow the shortest path between their nest and a food source. When searching for a food source, ants drop small amounts of pheromones so others can follow. When a food source is found, ants drop a significant larger amount of pheromones. When other ants leave the nest, the paths with larger amounts of pheromone have higher probability of being chosen. However, pheromone evaporates over time which results in larger amounts of pheromones for the shortest paths, as they are traveled by more ants.

The study of this complex behavior has resulted in many intelligent algorithms that have been successfully used for resolving complex and time consuming problems. The Ant Algorithm has been used to solve many different problems. This work will focus on the usage of an ant algorithm for resource searching in swarm-like P2P networks. By marking desired links with a synthetic pheromone, message forwarding will be done through the shortest path with higher probability, and thus save network resources such as bandwidth. Peers processing power can also be saved by processing less undesirable messages.

### ASAP

Although P2P networks are based on decentralized architectures, many still depend on a centralized server for resource search within the network. Despite data exchange being done between nodes, resource search is done outside the network and depends on centralized services. The objective

of this work is to study how this dependency could be broken by implementing in-network resource search through an ANT algorithm for query routing. Based on the ant foraging behavior, a distributed search engine can be implemented inside a P2P network. By marking, with synthetic pheromones nodes and connections that share same interests, a node can easily find the resource it is looking for, without having significant impact on the network's performance.

In order for a node to persist the routing information, each node needs three tables:

1. A table with pheromone values for each category, for each node. Each node can have different resources or know others with same, similar or very different resources. For this reason it is important to differentiate the nodes based on the category of the resources.
2. A table with an association between each category and its strategy weight value. Each node needs different strategies for processing different category searches, depending on the amount of information it has already gathered from previous searches for that category.
3. A table with the TTL value for each category. In order to improve network performance, the TTL value can be different for each category based on the information gathered by previous searches.

When a node searches for a specific resource in the network, it creates a query with a keyword and its respective category. The keyword can be any sequence of characters. As for the categories, they are predefined for the whole network and cannot be changed for a single node. Every resource has to be assigned to a category, otherwise it cannot be found.

When a node receives a query message, it checks if it holds any resource for the searched category. If the node is sharing resources for that category and it matches the searched keyword, then the node sends a reply message back through the same path. The reply message is responsible for updating the routing tables for each node it visits on the way to the one that started the query. This information will then be used by future searches for message routing optimization.

The algorithm is divided into three stages:

1. State transition: When ants choose the path to follow. Depends on the amount of pheromone the path has and the number of resources the end node has.
2. Pheromone update: When an ant finds what it is looking for, it drops an amount of pheromone that depends on the cost of the path.
3. Pheromone evaporation: Over time, pheromone evaporates and decreases for each of the marked paths.

The first step, state transition, is defined by two strategies: (1) exploration, where the ant explores the surrounding nodes and (2) exploitation, where the ant exploits the

existing information collected by its predecessors. The exploitation strategy selects the node, to which the message is sent, by choosing the one that has the highest combination of both pheromone value and number of resources. As for the exploration strategy, instead of choosing only the one with the highest combination of both values, it selects a set of nodes which have not yet been visited by the ant and calculates the probability distribution for all of them, based on both pheromone and number of resources values. If their probability is above a pre-determined threshold, the ant is cloned and sent to each one of these nodes. In order to select one of the strategies, the node uses the following equation:

$$\begin{cases} \text{Exploration,} & \text{if } rand > w_c \\ \text{Exploitation,} & \text{if } rand \leq w_c \end{cases} \quad (1)$$

where  $rand \in [0, 1]$  is a random generated number and  $w_c \in [0.2, 0.8]$ , a threshold to define the probability of each strategy being chosen for category  $c$ . This parameter starts at its minimum value of 0.2 and is incremented each time there is an update to the pheromone tables for a given category, to a maximum value of 0.8. When there is a pheromone evaporation, this value is decremented. This forces the algorithm to use an exploration strategy during the startup phase and, as the node gathers more and more information about its surroundings, the exploitation strategy is preferred, in an attempt to increase network performance. Equation 2 shows how this value is modified, where  $\tau_{update}$  and  $\tau_{evaporate}$  are weight values that determine how much the  $w_c$  parameter should increment and decrement over time.

$$\begin{cases} w_c = \tau_{update} \cdot w_c, & \text{for pheromone update} \\ w_c = \tau_{evaporate} \cdot w_c, & \text{for pheromone evaporation} \end{cases} \quad (2)$$

After choosing the strategy, if exploitation, the ant chooses its next node  $s$  through Equation 3.

$$s = MAX_{u \in U \cap u \notin s(F_q)} \left( \tau_{cu} \frac{\gamma_u}{10} \right), \quad (3)$$

where  $U$  is the set of all known and active nodes,  $s(F_q)$  is the set of nodes already visited by this ant and others that it might know of,  $\tau_{cu}$  is the amount of pheromone for category  $c$  in node  $u$ ,  $\gamma_u$  represents the total number of different resources node  $u$  has.

If the choice was exploration, then the node chooses a set of neighbors to send the ant to, instead of sending just to the one with the highest pheromone concentration and highest number of resources. First, the node calculates the probability distribution for all neighbor nodes, through Equation 4.

$$p_j = \frac{\tau_{cu} \frac{\gamma_u}{10}}{\sum_{u \in U \cap u \notin s(F_q)} \tau_{cu} \frac{\gamma_u}{10}}, \quad (4)$$

then, with the number of nodes already visited by the ant, a percentile  $q$  is calculated using Equation 5.

$$q = \begin{cases} 1, & \text{if } \beta * Count(F_q) \geq 1 \\ \beta, & \text{if } \beta * Count(F_q) = 0 \\ \beta * Count(F_q), & \text{Otherwise} \end{cases} \quad (5)$$

where  $\beta$  is a constant, such that  $\beta > 0$  and  $Count(F_q)$  is the total number of nodes that have been visited by this ant and others that it might know of. After this, all the probability distribution values, calculated in Equation 4, are sorted in ascending order where the value  $k$  corresponds to the percentile  $q$  for that list of values. For every node  $j$  that satisfies the condition  $p_j \geq k$ , the ant forwards its clone to that node.

The exploration strategy will gradually evolve into an exploitation strategy as the ant visits more and more nodes, being  $\beta$  the value that defines the speed at which this happens. Furthermore, in an exploration strategy, when the ant clones itself it passes all its information to its clones, including the nodes that will be visited by each of its clones. For example, node A send two cloned ants to node B and node C. The cloned ant that reached node B will know that there was another ant that was sent to node C and thus will not send new ants to that node as it has already processed the query. These methods are mainly used for improving network performance, since it is expected to result in a lower number of messages sent and processed by each node. These methods decrease the number of messages between two nodes that have already processed the message, thus decreasing redundant cross-communication. For identifying a message as corresponding to an already processed one, each message is associated with a unique identifier and nodes keep a record of all the messages they already processed. This way, nodes discard messages that have already been processed. In both strategies, the total number of different resources, the neighbor node has, is also used to calculate if an ant should be forwarded to that same neighbor node. The nodes with a higher number of resources are more likely to have a resource a node is looking for. Furthermore, since they participate in multiple swarms, the query will be sent to a higher number of swarms, increasing the probability of finding an answer. This is also increasing the speed of convergence of the algorithm to find the shortest path.

A message is also associated with a time-to-live (TTL). This parameter is used so that the query does not continue endlessly. If the TTL is reached during a search, the ant stops and dies and no result is found. However, if the ant finds a result, it stops and goes back the same path. In swarm-like networks, where nodes are grouped into swarms that share the same resources, you only need to find one node to be able to join the swarm(s) that node is participating in. For this reason, the ant does not need to continue after finding a result. When returning after finding a result, the ant updates all the pheromone values for the searched category in each node it passes through. The pheromone value is updated as shown in Equation 6.

$$\begin{cases} \tau_{cu} = \tau_{cu} + Z, \\ Z = \gamma_q \frac{TTL_{max}}{2 \cdot h_{qr}} \end{cases} \quad (6)$$

The pheromone update equation depends on: the number of nodes visited by the ant from the response node to the current node ( $h_{qr}$ ), the maximum TTL for the ant ( $TTL_{max}$ ) and the value  $\gamma_q$  which represents the total number of swarms the response node participates in. This equation was designed to differentiate the responses to a query based on the total amount of different resources the response node holds as well as the number of nodes between origin and destination. This means that a path to a node that has a higher number of resources will have a higher increment in the pheromone values than another node that, despite also having the desired resource, has a lower number of resources. The pheromone increment is also higher for smaller size paths, that is paths with a lower number of nodes between origin and destination.

As for the pheromone evaporation, a global update rule was used. For every predefined interval  $T$ , each node applies the pheromone evaporation rule shown in Equation 7 for every row in its routing table. In this rule,  $\tau \in [0, 1]$ , is the amount of pheromone that should evaporate in this interval.

$$\eta_u = \eta_u \cdot (1 - \tau) \quad (7)$$

Many of the parameters used in the algorithm should be tuned based on network properties. For example, parameters such as  $TTL_{max}$  or  $\beta$  need to take into account the network size and average swarm size, so that the search does not have a negative impact on both network performance and user experience.

### Use case

In order to fully understand the algorithm's workflow, we present a use case scenario where a node participating in a P2P network queries for a given resource. Figure 2 represents the network for this use case, where the lines represent the connection between two nodes. There are three swarms: A, B and C. While there are some nodes that only participate in a single swarm, nodes B1 and C1 participate in two swarms and work as a bridge between swarms A and B and swarms B and C, respectively. Each of the swarms represents a different resource. Node A1 just joined the network and has no information on its neighbors.

In this scenario, Node A1 initiates a query for the resource being shared in swarm C by Nodes C1, C2 and C3. First, the algorithm calculates  $rand$  value and determines the usage of exploration and, since it has no information on its neighbors, it sends an ant to both Node A2 and A3. Since both ants know that they were sent to all of Node A1's neighbors, no messages are exchanged between Node A2 and A3, since they are already processing the search query. Node A2

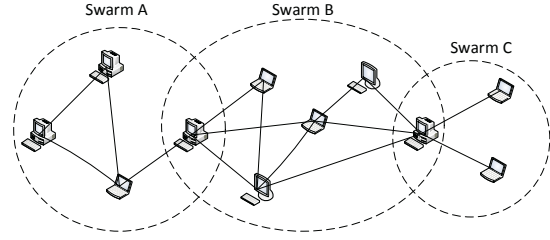


Figure 2: Swarm-like P2P network for use case.

does not have the content and does not have any other neighbors so it discards the message. As for Node A3, it uses the exploitation strategy and forwards the ant to Node B1. After this Node determined the use of exploration strategy, it calculated the following probability distribution values for its neighbors: [B3 - 90 ; B2 - 103 ; B4 - 250]. Then it calculates the percentil  $q = 0.5$  that, when applied to the probability distribution values, results in  $k = 103$ . Since both Nodes B2 and B4 have a value equal or greater than  $k$ , the ant is forwarded to these nodes and not to Node B3. In Node B1, the ant only passed through one node thus the percentil  $q$  is low. However, if the ant had already passed through a high number of nodes, the percentil value would be much higher and the ant would only be sent to Node B4. After receiving the ants, Nodes B2 and B4 don't exchange messages between each other since each one knows that the other is already processing that same search, avoiding sending redundant messages. After receiving the ant, Node B4 forwards it to Node B5 which forwards the ant to Node C1. Node B2 also forwards the ant to Node C1. In this case, Node C1 can actually receive the query two times since Node B2 and Node B5 don't share the same parent ant so they cannot know that the other one also sent an ant to Node C1. Since Node C1 has the desired resource, the search ants are not forwarded anymore and a response ant is sent through both paths. The response ants will increase the pheromone values for every link they pass through, following the paths: (1) C1-B2-B1-A3-A1 and (2) C1-B5-B4-B1-A3-A1. Despite the pheromone increase in both paths, path (1) will have a higher increase than path (2) since the ant passes through a fewer number of nodes. The paths A1-A3 and A3-B1 will have their pheromone value increased twice since both responses pass through these paths. This pheromone value is only increased for the category to which the desired resource belongs to. Future queries initiated by Node A1, for a resource with that same category, will have a higher probability of following the previously discovered paths.

### Simulation and Results

The simulated environment consisted of a content sharing swarm-like network topology with a total number of 10000

nodes randomly distributed over 50 swarms, according to the values in Table 1. Each swarm had a minimum size of 200 nodes and each node was connected to at least 10 other nodes. Each swarm was also associated with a different keyword that identified the content being shared on that same swarm. Since swarms are independent and unrelated with each other, some nodes participated in multiple swarms, working as a bridge between the different swarms. Table 1 shows the number of nodes that participated in multiple swarms.

Number of swarms	1	2	3	4	5	6
Number of nodes	2000	4000	2500	700	500	300

Table 1: Number of nodes that participate in one or more different swarms.

The experiment ran on a dynamic network, where a random number of nodes entered and left the network at any given time. However, no more than 5% of the network size (in this case, 500 nodes) could leave the network at the same time. Whenever the nodes reentered the network, they joined the same swarms they were previously participating in, since the node already had or was interested in those resources. However, the node established new connections different than the previous ones. The pheromone tables and other dynamic parameters, such as the strategy weight value were also set to default values upon reentering the network. As for the categories, we defined five: movie, music, application, tv show and book. During the experiment, 12000 queries were made.

Table 2 shows the initial parameter values. Since  $\beta$  defines the speed at which the algorithm evolves an ongoing search from an exploration strategy to an exploitation strategy, we varied this parameter in order to show how important its correct configuration is for the algorithm to achieve best performance.

Parameter	Value
Initial weight strategy for each category ( $w_c$ )	0.2
Weight strategy update ( $\tau_{update}$ )	1.25
Weight strategy evaporate ( $\tau_{evaporate}$ )	0.85
Initial pheromone value ( $\tau_{cu}$ )	0.009
Pheromone evaporation ( $\tau$ )	0.03
TTL	24
$\beta$	1; 0.1; 0.01; 0.006

Table 2: Algorithm parameters value.

As explained in Section IV-B, the strategy weight value

changed throughout the experiment, from a minimum value of 0.2 to a maximum value of 0.8. This allows a node to explore with higher probability its surroundings during the startup phase, when it enters the network and has no information whatsoever. This will lead to a higher number of messages exchanged during the startup phase. However, as the node gathers information about its neighbors, the preferred strategy changes to exploitation, which in turn will reduce the number of exchanged messages, as well as the paths an ant follows that do not lead to an answer, thus improving network performance. Figures 3 and 4 show this behavior.

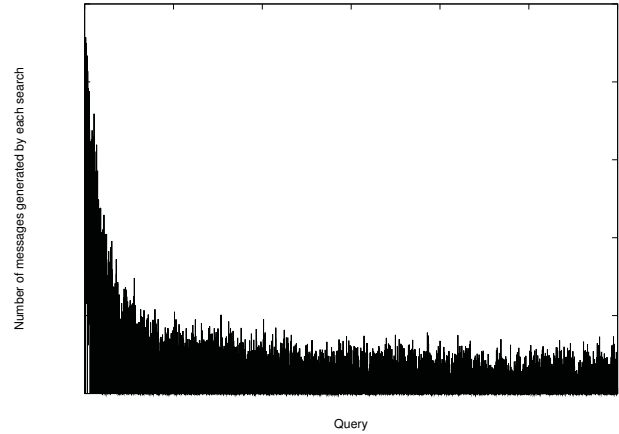


Figure 3: Number of messages generated by each query, with  $\beta = 0.01$ .

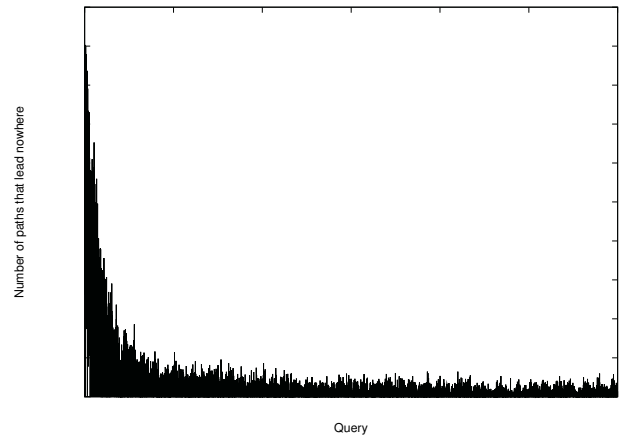


Figure 4: Number of paths that do not lead to an answer per query, with  $\beta = 0.01$ .

The first experiment ran with a  $\beta$  value of 0.01 and the algorithm's performance was compared to two modified versions of it, in order to show the importance of two choices made during the algorithm's development. In the first modified version (No parent memory), we modified the way the



cloned ant keeps the memory from its parent ant. When an ant cloned itself, it did not pass to its clone the information referring to all the node's neighbors that it would also be sent to. This way, an ant only knows the nodes that it visited and not the nodes visited by its first clones. As for the second modified version (No number of resources), the parameters  $\frac{\gamma_u}{10}$  (the total number of different resources node  $u$  has) were removed from the algorithm. This way, the exploration/exploitation strategy equations were modified to only take into account the pheromone value. The first measurement made was the average number of messages processed by each node. Table 3 shows the results.

Number of swarms node participates in	Original	No number of resources	No parent memory
One	49,56	1119,17	3212,34
Two	224,05	1750,16	6253,71
Three	1365,15	2603,32	9132,34
Four	3438,31	3636,62	12959,42
Five	6625,93	4875,91	17700,18
Six	11272,8	6398,05	23340,49

Table 3: Average number of messages processed by each node that participates in one or more swarms.

Network performance is achieved by reducing the number of messages generated by each search to a minimum value so that a resource is still discovered. Table 3 shows just that. By passing to the cloned ant the information about all the node's neighbor the other cloned ants are also going to be sent to, the ant has more information about which nodes have already processed the search, avoiding visiting again these nodes. This way, the number of messages in the network decreases without affecting the search algorithm performance. As for the removal of the  $\frac{\gamma_u}{10}$  parameter, it allows the algorithm to explore more the nodes with a fewer number of resources, thus distributing the workload more evenly in the network. However, the nodes with more resources are the ones that have a higher number of connections and have the highest probability of holding a resource or knowing someone that holds a resource that is desired. Although the usage of this parameter might have an impact on the network performance, as it does not distribute the workload evenly, it increases the probability of a node finding a resource in the network. Figure 5 shows the importance of the memory passing mechanism and differentiating nodes based on the number of different swarms they participate in, and how it can affect user experience when searching for resources in the network.

Both Table 3 and Figure 5 also show the tradeoff there needs to be between network performance and search performance. When distributing more evenly the workload in the network, we also decrement the number of queries that find

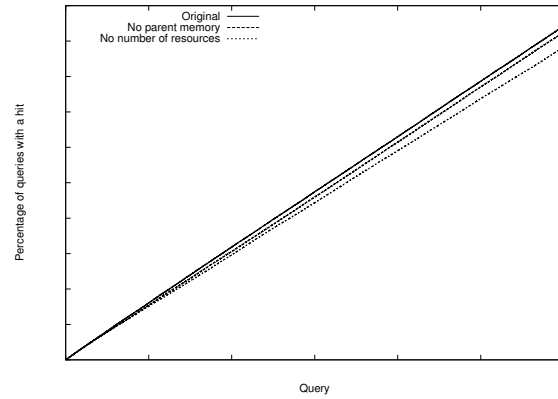


Figure 5: Cumulative distribution function for the queries with a hit.

at least one result. When focusing the workload in the nodes that have a higher probability of having a resource, we increment the number of queries that find the search resource, at the expense of network performance. Figure 5 also shows an unexpected behavior. When removing the memory mechanism, the algorithm generates a much higher number of messages (Table 3), however it achieves a lower hit rate than the original version. This can be explained by the large number of generated messages that are sent to nodes that have already processed the message, and thus are discarded.

After showing the importance of these algorithm's behaviors, we ran the simulation with four different  $\beta$  values: 1, 0.1, 0.01 and 0.006<sup>1</sup>. The  $\beta$  parameter defines how the exploration strategy evolves into an exploitation one, having direct impact on both network and algorithm performance. First, we compared how this value affects both the distribution of the workload in the network and the average number of messages generated by each search. Table 4 shows the results.

Number of swarms node participates in	$\beta$			
	1	0.1	0.01	0.006
One	1,63	29,29	49,56	78,76
Two	1,88	51,2	224,05	720,54
Three	2,25	64,67	1365,15	3853,47
Four	2,7	96,32	3438,31	7322,91
Five	63,25	656,66	6625,93	11798,08
Six	781,49	4954,7	11272,8	16987,04

Table 4: Average number of messages processed by each node that participates in one or more swarms.

As expected, by increasing the  $\beta$  value, the exploration strategy evolves more rapidly into an exploitation strategy,

<sup>1</sup> Simulator limitations forced the lowest value to be 0.006

generating a lower number of messages and focusing most of these messages in the nodes with the higher number of resources. As the value decreases, the algorithm makes a larger use of the exploration strategy, thus generating more messages and exploring more the nodes with a lower number of resources. This exploration strategy is very important in gathering information from the surrounding network so that more paths can be explored and the desired resource found. This has a direct impact on the search results and consequently the algorithm's performance. However, having a  $\beta$  value too low will result in a network flooding and consequently have a negative impact on network performance. Figure 6 shows a CDF with the percentage of queries that find the searched resource.

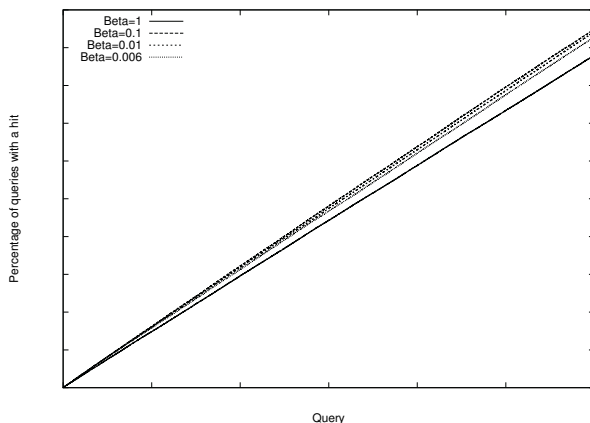


Figure 6: CDF for the queries with a hit, with different values for the  $\beta$  parameter.

In order to configure the  $\beta$  parameter, the average swarm size in the network and the percentage of nodes that actually participate in multiple swarms need to be taken into account. If a network has a low percentage of multi-swarm nodes, the  $\beta$  value should be low enough so that the algorithm does not focus all its messages on these nodes and distributes its load throughout all other nodes. This behavior can be observed in Table 6 for  $\beta = 1$ . On the other hand, if the network has a high percentage of multi-swarm nodes, the  $\beta$  value should be high enough so that it does not flood the network with query messages. The  $\beta$  parameter can have a significant impact, either positive or negative, on network and algorithm performance and thus should be configured accordingly to each network it is used in.

## Conclusions

This work presents an ant-based algorithm capable of providing a search mechanism inside swarm-like P2P networks. The algorithm focuses on minimizing the impact it has on network performance without affecting the user experience. This is done through pheromone marking, strategy and message forwarding adaptation, based on the total number of

nodes already visited by the search ant. This way, the number of messages generated by each search query decreases as the node has more information about the surrounding network. Our simulations show how the algorithm behaves and reacts to different parameter configuration and how it can be configured to achieve best network performance and results. The simulation results also show the reason for some design and development choices made upon creation.

## Acknowledgements

This work was supported by national funds through FCT - Fundação para a Ciência e a Tecnologia, under project PEst-OE/EEI/LA0021/2013. The authors would also like to thank Professors Ricardo Pereira and Fernando Mira da Silva for their support.

## References

- Chen, Z., Liu, J., and Li, J. (2010). An adaptive expanding antbudget search algorithm for unstructured p2p networks. In *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, volume 2, pages V2-777 –V2-781.
- Cisco (2011). Cisco visual networking index: Forecast and methodology, 2010-2015.
- Dorigo, M. and Caro, G. D. (1999). Ant colony optimization: A new meta-heuristic. In *Proceedings of the IEEE Congress on Evolutionary Computation*.
- Dorigo, M. and Gambardella, L. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. In *IEEE Transactions on Evolutionary Computation*.
- Gkantsidis, C., Mihail, M., and Saberi, A. (2005). Hybrid search schemes for unstructured peer-to-peer networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1526 – 1537 vol. 3.
- Lv, Q., Cao, P., Cohen, E., Li, K., and Shenker, S. (2002). Search and replication in unstructured peer-to-peer networks. In *Proceedings of the 16th international conference on Supercomputing, ICS '02*, pages 84–95, New York, NY, USA. ACM.
- Michlmayr, E. (2006). Ant algorithms for search in unstructured peer-to-peer networks. In *IN PROCEEDINGS OF THE PH.D. WORKSHOP, 22ND INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE)*.
- Montresor, A. and Jelasity, M. (2009). PeerSim: A scalable P2P simulator. In *Proc. of the 9th Int. Conference on Peer-to-Peer (P2P'09)*.
- Sandvine (2011). Global internet phenomena report.
- Wu, C.-J., Yang, K.-H., and Ho, J.-M. (2006). Antsearch: An ant search algorithm in unstructured peer-to-peer networks. *2012 IEEE Symposium on Computers and Communications (ISCC)*, 0:429–434.
- Wu, G., Liu, J., Shen, X., Gao, L., Xu, J., and Xi, K. (2008). Erant-budget: A search algorithm in unstructured p2p networks. In *Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on*, volume 2, pages 765 –769.

# Population dynamics with limited perception establish global swarm topology

Anna Shcherbacheva<sup>1</sup>, Tuomo Kauranne<sup>1</sup>

<sup>1</sup>Lappeenranta University of Technology, Finland  
anna.shcherbacheva@lut.fi  
tuomo.kauranne@lut.fi

## Abstract

We simulate the swarming behavior of three synthetic animal species that differ only by the degree of perception they have on their fellow animals. The species are called *mosquitoes*, *birds* and *fish*. The swarms that comprise many individuals of each species in turn move randomly in a rugged potential landscape. The *mosquitoes* pay no heed to one another. The *birds* follow a bunch of their nearest neighbours in front, based on strictly limited visibility. The *fish*, in turn, sense also far-away neighbors through their lateral line, as modeled by an exponentially decaying perception function. The simulations show that such local differences in perception by swarming individuals have global macroscopic consequences to the geometry of the corresponding swarms. These consequences are of persistent nature across many simulations with each species.

## Introduction

Humans, like many other animal species, are social. We are fundamentally geared to living in a herd of some 20 - 100 individuals, from our nearest primate cousins to decide. Many other animal species have much more intense social lives, with flocks extending to thousands of individuals. Large flocks - or swarms - have their own requirements as to the means of imposing collective social control over the individuals involved. The process of social co-ordination is bi-directional: on the one hand, swarm dynamics exerts control over each individual. On the other, swarm dynamics is a direct consequence of the collective motion of all of its individuals.

In this article, we compare the consequences on collective dynamics of different degrees and forms of perception of swarming animals through computer simulations. Many studies indicate that the bi-directional flow of information described above has an important defining role in determining the nature of swarm dynamics. The impact of information flow boils down to the question of how do the individuals in a swarm perceive the collective dynamics of the swarm - and to the reciprocal question of how does the reaction of individuals influence the collective dynamics of the swarm.

We shall study this question with computer simulations of the collective motion of swarms of three different types

of animals, all capable of moving in two spatial dimensions. These synthetic species are labeled as *mosquitoes*, *birds* and *fish*. They are set to move in a similar synthetic world with some external forces and constraints. But the way they perceive their fellow passengers is different, which has a fundamental impact on the nature of the corresponding collective motion of the swarm.

## Swarm dynamics with a difference

Collective swarm dynamics can be described in many ways, such as using ordinary or partial, deterministic or stochastic, differential equations. Classical models of *Eulerian type* (e.g., see Milewski and Yang (2008), Murray (2002), Mogilner and Edelstein-Keshet (1999) and Nagai and M. (1983)) are based on the **diffusion-advection-reaction** equation, governing the spatio-temporal dynamics of the population density:

$$\frac{\partial f}{\partial t} = \frac{\partial}{\partial x} \left( D(f) \frac{\partial f}{\partial x} \right) - \frac{\partial}{\partial x} (V(f)f) + B(f), \quad (1)$$

where the first term on the right-hand side introduces a Brownian motion with diffusion coefficient  $D(f)$ , the second term stands for advection with density-dependent velocity  $V(f)$  and the last reaction term may include birth or death processes.

Convection term results in attraction and repulsion effects, reflecting forms of social interaction between population members which implies that the direction and speed of motion of a particular individual is determined by the population density of the surrounding environment. One advantage of continuous models is the diversity of readily available analytic tools that facilitate their study.

Since the sensory systems of animals are limited, it is typically assumed that interactions have finite spatial influence. In most PDE-based models advection velocity is specified as a convolution (Mogilner and Edelstein-Keshet (1999), Edelstein-Keshet et al. (1997)):

$$V(f) = \mathbf{K} * f = \int_R \mathbf{K}(x - x') f(x', t) dx', \quad (2)$$

where the kernel  $K$  relates to the strength of animal-animal interaction per unit density with a distance  $x - x'$  between two sites, see Mogilner and Edelstein-Keshet (1999).

Another common approach is based on modeling the movement of each individual member of the total population comprising  $N$  identical members. In this so-called *Lagrangian approach* each individual member follows simple rules of motion, specified by either a system of stochastic differential equations, as was done in Burger et al. (2007), Morale et al. (2005), or via a hierarchical algorithm with a probabilistic decision-making mechanism, see Gueron et al. (1996).

In our simulations of synthetic animal swarms, swarming *mosquitoes* do not affect each other. The main factor that determines their motion is sensing features of a potential prey or heat source. Host-seeking behavior of mosquitoes was thoroughly considered in Cummins et al. (2012). *Mosquitoes* were treated as a number of independent agents, sampling concentration of attractive odor emitted from host individuals via mechanism of *klinotaxis*. The concentration is given by the convection-diffusion equation. In the present study, the *mosquito* case is modeled with a similar approach, but by different means.

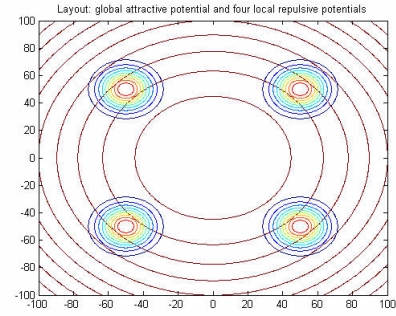
In order to focus our study on the influence of perception on collective dynamics, we use an identical environment for all the swarms. This environment consists of a two-dimensional landscape that resides at the bottom of a "potential bowl" that describes the local habitat of the swarm. The swarm is therefore confined to a limited, but boundaryless, area. Inside the potential bowl, there are four obstacles at the corners of a square that repel individuals and cause the swarm to turn away from them. These obstacles can represent e.g. trees, buildings, rocks or repellents.

The dynamics of a swarm are directed both by external forces - such as the potential bowl above that forms a basin of attraction, and the obstacles that each constitute a repulsive force - and by internal forces created by the autonomous dynamics generated by the members of the swarm through their interaction. Both external and internal forces are represented in population dynamics as force potentials that are mediated by perception. In our model, we assume the external forces to be constant in time and independent of swarm dynamics, so as to focus on the autonomous dynamics of the swarm. In the real world, both external and internal forces are very much dynamic, as can be seen e.g. from any movie of dolphins chasing a school of fish.

Our autonomous swarm dynamics are broadly following a class of stochastic differential equation models introduced by Capasso and Morale in Morale et al. (2005) and Burger et al. (2007). In this kind of dynamics, there is a long-range herding force that keeps the swarm together, and a short-range repulsion force that prevents individuals from colliding.

Our three fundamental species differ only in the way that

Figure 1: General layout: global attractive potential in the center and four local sources of repulsion.



they can perceive their fellow animals. All the three species are only synthetic forms of life, but they are each given a familiar name that links each synthetic species to a related animal species.

- *Mosquitoes* do not perceive their fellow mosquitoes. Instead they move in Brownian motion, weighted by the gradient of the external potential field alone.
- *Birds* do see their fellow birds, but only up to a fixed distance limit in their front.
- *Fish* may see only their immediate neighbors, but because of their special pressure sensing organ, the lateral line, they can feel the movement of the swarm beyond their range of visibility. In this case, their perception function is a radial, exponentially decaying function.

### Basic dynamic equations

We consider the general layout for the three above-mentioned cases, see Figure (1), where a source of global attractive potential is located at the center of the experimental habitat of the species, and four short-term repulsive potentials are placed separately from one another. In the case of a *mosquito* swarm, we treat this setting as an indoor space amended with four sources of repellent and one intensive heat source at the center. In the two other cases, the layout can be treated as a domain inhabited with a population of animals which contains four obstacles, such as trees or rocks.

#### Mosquito swarm

The first case examines collective behavior of *mosquitoes* in presence of an intensive heat source and four patches covered with repellent (e.g., spray), modeled as a system of non-interacting particles driven by an external potential.

The heat source at  $\mathbf{x}^h$  is modeled as a Gaussian hump:

$$C_a(\mathbf{x}) = \exp \left[ -\frac{d^2(\mathbf{x}, \mathbf{x}^h)}{2\sigma_a^2} \right], \quad (3)$$



where  $d(\mathbf{x}^h, \mathbf{x})$  is the distance between point  $\mathbf{x}$  and the position of the source  $\mathbf{x}^h$ . The standard deviation of the Gaussian  $\sigma_a$  conditions the minimal distance at which the *mosquito* is able to sense the heat.

We assume that each *mosquito* is driven towards the source by the mechanism of **klinotaxis**, as it was conjectured in Vickers (2000) for the case of host-seeking behavior of real mosquitoes. During klinotaxis, an animal samples the concentration of attractive substance at one location, then changes location and repeats sampling, using its memory of the concentration to choose its next position Cardé (1996).

In the present study, the klinotaxis mechanism is modeled by employing a random walk. The **Metropolis algorithm** was introduced in the 1950s in statistical physics literature as a tool to sample probability distributions, see Metropolis et al. (1953). The Metropolis algorithm is based on an **accept-reject** step. Assume that we take a step from point  $\mathbf{x}^{n-1}$  to a candidate position  $\mathbf{x}^n$ . If corresponding probabilities are  $p_{n-1}, p_n$ , we accept the new position with probability

$$\alpha_a = \min\left(1, \frac{p_n}{p_{n-1}}\right) \quad (4)$$

Hence, the upward steps ( $p_n > p_{n-1}$ ) are always accepted, while steps downwards are accepted with probability  $p_n/p_{n-1}$ . Probabilities  $p_n, p_{n-1}$  are taken from a **proposal distribution**. Applying the above-specified sampling rule with adequate proposal distribution implies that the samples converge towards the underlying target distribution.

In our simulations random walk is employed as a proposal distribution. Consider a *mosquito* at position  $\mathbf{x}_{n-1}$  at iteration  $n - 1$ . It randomly selects a candidate position  $\mathbf{x}^n$  by

$$\mathbf{x}^n = \mathbf{x}^{n-1} + d\mathbf{W}, \quad d\mathbf{W} \sim N(0, \Sigma), \quad (5)$$

where the two-dimensional Gaussian  $N(0, \Sigma)$  conditions the step length for the random walk. The probabilities are associated with the distribution of heat, and may be given as an exponential of the heat concentration:  $p_n = \exp(-C_a(\mathbf{x}^n)/2\sigma_a^2)$ . The accept/reject probability can then be written as follows:

$$\alpha_a = \min\left(1, \frac{p_n}{p_{n-1}}\right) = \min\left(1, \exp\left[-(C_a(\mathbf{x}^{n-1}) - C_a(\mathbf{x}^n))/2\sigma^2\right]\right), \quad (6)$$

where  $\sigma$  is a scale which governs the probability of a step away from the source to get accepted: the smaller is  $\sigma$ , the less likely they are to get accepted. Our *mosquito* 'sampler' is eventually supposed to roam in proximity to the heat source.

Repellent is treated as Heaviside step function which stands for a **probability of rejection** for candidate position  $\mathbf{x}$ :

$$\alpha_r(\mathbf{x}, \mathbf{x}^r) = \begin{cases} 1, & \min_{j \in 1, \dots, N_r} d(\mathbf{x}, \mathbf{x}^r) \leq L \\ 0, & \min_{j \in 1, \dots, N_r} d(\mathbf{x}, \mathbf{x}^r) > L \end{cases}, \quad (7)$$

where  $d(\mathbf{x}, \mathbf{x}^r)$  is a distance from position  $\mathbf{x}$  to the source of repellent,  $L$  determines the range of coverage. One way to combine multiple repellents is to sum rates of rejection over all sources of protection and take Metropolis-type of probability:

$$\alpha_r(\mathbf{x}^n) = \min\left(1, \sum_{i=1}^{N_r} \alpha_r(\mathbf{x}, \mathbf{x}_i^r)\right), \quad (8)$$

where  $N_r$  is a total number of repellents. A *mosquito* swarm is represented as a number of individuals placed initially at random spatial positions on a rectangular patch  $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$ . After that, every *mosquito* with initial position  $\mathbf{x}^0$  changes its location in accordance with the following algorithm:

#### Algorithm for swarm dynamics

1. Select a candidate position  $\mathbf{x}^n$  by adding Brownian increment to previous point, that is compute  $\mathbf{x}^n$  by formula (5).
2. Measure concentration of heat at new position as it was specified in formula (3)
3. Compute **probability of acceptance** for position  $\mathbf{x}^n$ :

$$\alpha_a(\mathbf{x}^{n-1}, \mathbf{x}^n) = \min\left(1, \exp\left[-(C_a(\mathbf{x}^{n-1}) - C_a(\mathbf{x}^n))/2\sigma^2\right]\right), \quad (9)$$

4. Compute **probability of rejection** by formula (8)
5. Generate a random number  $r \sim U[0, 1]$ ; if  $r < \alpha_a(1 - \alpha_r)$ , accept the new position  $\mathbf{x}^n$ . Otherwise, remain at the old position:  $\mathbf{x}^n = \mathbf{x}^{n-1}$ ;
6. Move to step 1,  $n \rightarrow n + 1$ .

#### Flocks of birds

Our second type of swarming behavior also features dynamics under a global attractive potential. It can be viewed as a population moving inside a wide domain of habitation. This domain includes several areas covered with obstacles that are avoided by population members. External attraction and repulsion is treated as in the previous case of *mosquito* swarming behavior.

In contrast to *mosquitoes*, this second type of animals is supposed to coordinate with fellow-individuals, i.e. exhibit a tendency to aggregate but to avoid over-crowding. In our present simulations, both interactive effects are introduced by means of **Kalman dynamics**, a term derived from the statistical data assimilation method known as **The Kalman Filter**, see Kalman (1960), Evensen (2003).

The Kalman filter produces a **state estimate** of a dynamic system as a weighted average of a **prior state** or predicted state, and of a **state observation**.

Suppose that at the step  $n - 1$  *birds* have occupied positions

$$\mathbf{x}^{n-1} = (\mathbf{x}_1^{n-1}, \mathbf{x}_2^{n-1}, \dots, \mathbf{x}_N^{n-1}), \quad (10)$$

where  $N$  is the number of *birds*. Firstly, **prior candidate positions**  $\mathbf{x}_i^n = \mathbf{x}_i^{n-1} + \mathbf{dW}$ ,  $i = 1, \dots, N$  are randomly selected and independent of one another. After that, every prior candidate point  $\mathbf{x}_i^n$ ,  $i = 1, \dots, N$  is amended with an observational increment:

$$\bar{\mathbf{x}}_i^n = \mathbf{x}_i^n + G_a (\mathbf{y}_i^a - \mathbf{x}_i^n) \quad (11)$$

to introduce cohesion towards closest individuals, which is a typical way of attaining synchronization between animals, such as schooling fishes and flocking birds.

In the current case, we assume that a long-range attraction rule applies to the nearest five neighbors of each individual, as has been observed in the case of real birds. Positions of viewed fellow *birds* are combined into an artificial state observation

$$\mathbf{y}_i^a = \frac{1}{N_o} \sum_{j=j_1}^{j_{N_o}} \mathbf{x}_j^n, \quad (12)$$

where  $j_1, \dots, j_{N_o}$  are indices of  $N_o$  closest neighbors to the side and ahead of the agent  $\mathbf{x}_i^n$ .

The **Kalman Gain**  $G_a$  can be adjusted to achieve a particular strength of cohesion: increasing  $G_a$  implies enhancement of aggregative behavior.

Short-range repulsion between *birds* can be modeled similarly:

$$\bar{\mathbf{x}}_i^n = \mathbf{x}_i^n - G_r (\mathbf{y}_i^r - \mathbf{x}_i^n), \quad (13)$$

where observation  $\mathbf{y}$  is composed as an average over the set  $\mathcal{N}_r = \{j | d(\mathbf{x}_i^n, \mathbf{x}_j^n) < d_{\min}\}$  of all positions located closer than at minimum distance  $d_{\min}$ :

$$\mathbf{y}_i^r = \frac{1}{|\mathcal{N}_r|} \sum_{j \in \mathcal{N}_r} \mathbf{x}_j^n. \quad (14)$$

Analogously to the above-described case of cohesion, the Kalman gain  $G_r$  in formula (13) governs the strength of repulsive interaction.

**Algorithm for collective dynamics** is applied to each individual agent  $i = 1, \dots, N$  separately

1. Select a candidate position randomly:

$$\mathbf{x}_i^n = \mathbf{x}_i^{n-1} + \mathbf{dW}, \mathbf{dW} \sim N(0, \Sigma). \quad (15)$$

2. Compute **probability of acceptance** for position  $\mathbf{x}_i^n$ :

$$\alpha_a = \min \left( 1, e^{-\frac{(C_a(\mathbf{x}_i^{n-1}) - C_a(\mathbf{x}_i^n))^2}{2\sigma^2}} \right) \quad (16)$$

3. Compute **probability of rejection** for position  $\mathbf{x}_i^n$ :

$$\alpha_r = \begin{cases} 1, & \min_{j \in 1, \dots, N_r} d(\mathbf{x}_i^n, \mathbf{x}_j^r) \leq L \\ 0, & \min_{j \in 1, \dots, N_r} d(\mathbf{x}_i^n, \mathbf{x}_j^r) > L \end{cases}, \quad (17)$$

where  $d(\mathbf{x}_i^n, \mathbf{x}_j^r)$  is a distance between the  $i$ -th agent and the center of  $j$ -th obstruct patch,  $L$  stands for the width of the patch,

4. Accept new position  $\mathbf{x}_i^n$  with probability  $\alpha_a(1 - \alpha_r)$ , in case of rejection, stay at the old position  $\mathbf{x}_i^n = \mathbf{x}_i^{n-1}$ ,
5. Apply Kalman dynamics to induce interactive behavior (cohesion and repulsion):

$$\begin{aligned} \bar{\mathbf{x}}_i^n &= \mathbf{x}_i^n + G_a (\mathbf{y}_i^a - \mathbf{x}_i^n) \\ &\quad - G_r (\mathbf{y}_i^r - \mathbf{x}_i^n), \end{aligned} \quad (18)$$

where observations  $\mathbf{y}_i^r$  and  $\mathbf{y}_i^a$  are computed by formulas (12) and (14), correspondingly;

6. Compute probability of rejection  $\alpha'_r$  for position  $\bar{\mathbf{x}}_i^n$  by formula specified in the item 3, accept position ( $\mathbf{x}_i^n = \bar{\mathbf{x}}_i^n$ ) with probability  $1 - \alpha'_r$ , otherwise, remain at the old position  $\mathbf{x}_i^n$ ,
7. Move to step 1,  $n \rightarrow n + 1$ .

### Schools of fish

Schooling behavior is modeled similarly to the previous case, except for the observation stage for cohesion. *Fish* are characterized by their ability to sense their fellow kin even beyond a certain range, which is reflected in the perception function applied at the state observation step as follows:

$$\mathbf{y}_i^n = \sum_{\substack{j=1 \\ j \neq i}}^N \exp[-\lambda d(\mathbf{x}_i^n, \mathbf{x}_j^n)] \mathbf{x}_j^n, \quad (19)$$

where observation weights decay exponentially with distance.

### Results

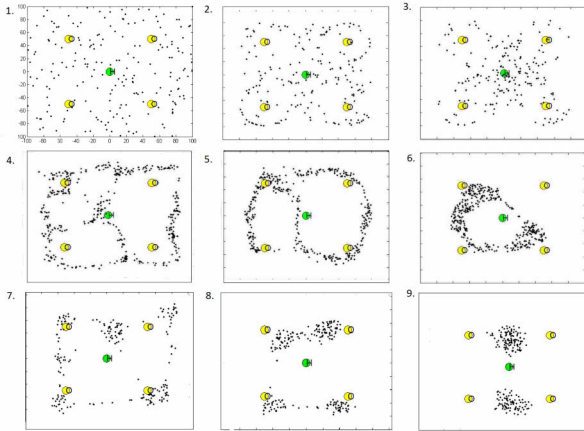
As to the pun in the title of the article - *mosquitoes* do not recognize any friends. *Mosquitoes*, by their definition as Markovian synthetic animals without regard for their fellow kin, keep happily flying on top of one another. It is lucky that they can be regarded as point-like particles in simulations!

The results of the simulations for all three species are illustrated in Figure 2. The three rows of the figure represent the three species. The columns from left to right represent an early state, one or two intermediate states, and a late state of the swarm, respectively, in each case. The upper-most left figure represents the common random initial state of all

Table 1: Values of model coefficients employed in experimental runs.

parameters	values
$\sigma_a$	1000
$L$	4
$G_a$	0.7
$G_r$	0.9
$\lambda$	1
$\sigma$	$1e - 3$

Figure 2: 1.Initial layout, 2.*Mosquito* swarm distribution at an intermediate stage, 3.The final distribution of the *mosquito* swarm, 4.-5. Intermediate distributions of *birds*, 6. The final distribution of *birds* ,7.-8. Intermediate distributions of *fish* , 9. The final distribution of *fish*



three swarms. From these figures, we conclude that the answer to the pun is not the same for *birds* and *fish* as it is for *mosquitoes*. The former two species both react to their fellow flyers or swimmers, in a manner that has a fundamental impact on the geometry of the swarms they form. Despite the slightly different forms of the algorithms used for *mosquitoes* and for the other two "species" of synthetic creatures, their swarms start out similar. The disregard for fellow *mosquitoes* causes them to get distributed according to the geometry of the landscape, but concentrating on the level contours of their potential well. Only in late stages of the simulation do *mosquitoes* finally find their way to the bottom of their potential bowl.

*Birds*, on the other hand, are so attached to their fellow fliers that this property actually prevents them from reaching the bottom of the potential bowl. Instead, they circulate in a simply or multiply connected chain to the foreseeable future. This is caused by the attractive potential of the nearest *birds* that inexorably pull the *birds* away from their preferred feeding ground, because of the "horror vacui" that surrounds

it.

Finally, *fish* display yet another variety of swarm dynamics, since they can sense the presence of also far-away fellow swimmers. This property allows them to cluster into groups, instead of a continuous chain. In our rugged landscape, the end result is not a single cluster, but two clusters of *fish*, held away from each other and from the best feeding ground by the balance of mutual repulsion that would strengthen if the two schools would move closer to one another.

In all three cases, the final distributions of the swarms, flocks and schools are quite stable over many cycles of simulations. They therefore demonstrate clearly the macroscopic geometric consequences of the local degree of perception attributed to each synthetic species.

Increasing Kalman gain  $G_a$  induces clustering behavior in the swarm, hence the latter parameter should be kept sufficiently big to stimulate cohesion, but not exceedingly large, since it annihilates the motion caused by an external potential. To avoid this effect, Kalman gain which governs the attraction should be smaller than matching repulsion coefficient  $G_r$ , see Table (1).

## Discussion

We have demonstrated that under similar but non-homogeneous geometric circumstances, the degree of local perception by swarming animals has global geometric consequences for their corresponding swarm dynamics. Instead of behaving in a totally random or chaotic fashion, the swarms adopt persistent geometric shapes that are a function of the degree of local perception possessed by the individuals in the swarm, even as the individuals generally move in a similar stochastic manner. We have demonstrated three such geometries in the case of three synthetic species that behave like mosquitoes, birds and fish, respectively.

This empirical result is not very surprising, in view of corresponding results in deterministic differential geometry. In differential geometry, the difference between the dimensions of the kernel and co-kernel and of a local linear curvature operator, such as the Laplacian, determines uniquely the Euler characteristic of the manifold, as is stated by the Atiyah-Singer Index Theorem and its many analogues and generalizations. The Euler characteristic describes the difference between the number of edges and the number of vertices of an arbitrary triangulation, or simplicial complexification in dimensions higher than two, of the corresponding manifold and it is a topological invariant. This means that the Euler characteristic remains the same, no matter how the triangulation or simplicial complex have been constructed. It is, in particular, independent of the length of the edges in such a triangulation, and hence on its spatial resolution.

On the other hand, there are well-known analogies between curvature operators and random walks, such as the bijective relationship between Brownian motion and the

Laplacian. It is not trivial to extend such results, that depend crucially on the linearity of the curvature operator, to the non-linear stochastic dynamics discussed in the current paper. But the persistence of the limit geometry of the swarms observed in our numerical experiments seems to indicate that the bridge of analogies from random walks through corresponding differential dynamics onto the global topology of the limit swarm on a non-trivially connected manifold is a continuous path. But the fact that the resulting topology is different for different perception operators testifies to the non-trivial nature of these analogies: the topology of the swarm is not uniquely determined by the topology of the underlying spatial manifold, but also depends on the non-linear perception operator associated with the swarm dynamics.

## References

- Burger, M., Capasso, V., and Morale, D. (2007). On an aggregation model with long and short range interactions. *Nonlinear Analysis: Real World Applications*, 3:939–958.
- Cardé, R. (1996). Odour plumes and odour-mediate flight in insects. In *Olfaction in mosquito-host interactions.*, number 200 in Ciba Foundation Symposium, page 54–66. John Wiley and Sons Ltd.
- Cummins, B., Cortez, R., Foppa, I., Walbeck, J., and Hyman, J. (2012). A spatial model of mosquito host-seeking behavior. *PLoS Comput Biol*, 8(2):e1002500.
- Edelstein-Keshet, L., Watmough, J., and Grunbaum, D. (1997). Do travelling band solutions describe cohesive swarms? an investigation for migratory locusts. *Journal of Mathematical Biology*, 36:515–549.
- Evensen, G. (2003). The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367.
- Gueron, S., Levin, S., and Rubenstein, D. (1996). The dynamics of herds: from individuals to aggregations. *Journal of Theoretical Biology*, 182:85–98.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092.
- Milewski, P. A. and Yang, X. (2008). A simple model for biological aggregation with asymmetric sensing. *Communications in Mathematical Sciences*, 6(2):397–416.
- Mogilner, A. and Edelstein-Keshet, L. (1999). A non-local model for a swarm. *Journal of Mathematical Biology*, 38:534–570.
- Morale, D., Capasso, V., and Oelschläger, K. (2005). An interacting particle system modelling aggregation behavior: from individuals to populations. *Journal of Mathematical Biology*, 50(1):49–66.
- Murray, J. (2002). *Mathematical Biology I. An Introduction*. Springer.
- Nagai, T. and M., M. (1983). Some nonlinear degenerate diffusion equations related to population dynamics. *The Journal of the Mathematical Society of Japan*, 35:539–561.
- Vickers, N. (2000). Mechanisms of animal navigation in odor plumes. *Biological Bulletin*, 198:203–212.



# Some Remarks on Dynamics of Binary Chromosomes Population

Zbigniew Pliszka<sup>1</sup>, Olgierd Unold<sup>2</sup>

<sup>1</sup>Wroclaw Public Library, Sztabowa 95, 53-310 Wroclaw, Poland

<sup>2</sup>Wroclaw University of Technology, Wyb. Wyspianskiego 25, 50-370 Wroclaw, Poland  
olgierd.unold@pwr.wroc.pl

Population is the fundamental basis of any evolution (Novak, 2006). This statement is true both in biological and artificial evolutionary systems. Population dynamics studies short and long-term changes of the certain population features, and is a branch of life sciences. Here, we discuss new properties of population of binary chromosomes evolved by genetic algorithm (GA), which is one of the possible implementations of artificial evolutionary systems. In particular, we introduce a theorem (Theorem 3) allowing us to determine the minimal number of simple operations necessary to restore the entire space to explore. The work is partly based on Pliszka and Unold (2011, 2012).

The discipline of GAs (and broadly Evolutionary Computation EC) is still focused more on the empirical aspects of algorithms than theoretical studies. Methods, which are currently in use in theoretical studies of these algorithms, could be classified into one of the following groups: schema theory, Markov chains theory, dimensional analysis, order statistics, quantitative genetics, orthogonal functions analysis, quadratical dynamical systems, and statistical physics. Simplistic assumptions have frequently been adopted in the theoretical analyses and these have deformed the analyzed algorithms in such a way that they question the real connection between the results obtained and the investigated algorithms.

The essential step in GA/EC is to determine the representation of computational population (Hu and Banzhaf, 2010). Another one is to define the method of gene duplication. Most of GAs use linear binary representations, and the most standard one is an array of bits. Due to fixed size of such representations, their parts are easily aligned. This facilitates simple crossover operation.

In our approach binary chromosomes are represented as binary, fixed-length chromosomes, using an alternative to zero-one decoding technique, called Hadamard representation. The search space  $\{0, 1\}^n$  was replaced by  $\{-1, 1\}^n$ . Thanks to use a new binary model the requirement of orthogonal columns pairs is omitted. Subject of this study is the following set:

$$\begin{aligned} H^n &= \{(h_{s,n}, h_{s,n-1}, \dots, h_{s,2}, h_{s,1}) : \\ &\forall s \in \{0, 1, \dots, 2^n - 1\} \\ &\forall i \in \{1, 2, \dots, n\} \\ &h_{s,i} \in \{-1, 1\}\} \end{aligned}$$

Its elements represent all possible binary chromosomes of equal length  $n$ , where  $n$  is a natural number higher than 1. Note that the Hadamard representation is in fact a transformed Hamming space. The proposed representation has one, apparently insignificant property, which distinguishes it from the binary representation: a square of each coordinates is equalled 1. This fact draws two subsequent conclusions: the sum of the squares of coordinate of each element of the  $H^n$  space is constant and equals this space dimension, and there is no element with zero coordinates. The collection of these simple facts allows for the formulation of rules for phenotypes (indices) and development of automate methods of moving frame  $H^n$ , as well as determination of the distance (level of differentiation) between the elements of this space (Pliszka and Unold, 2011).

The use of Hadamard representation allows us to give theoretical proof for epistatic properties as well as exploration possibilities of a crossover operator. We say that the population is ancestral, if all its elements can be obtained from a primary (initial) population as a result of the assembling only crossing-overs.

**Theorem 1.** The whole space  $H^n$  is the ancestral population if and only if there are the elements in the primary population  $P$ , which have the following properties: for each locus, we have two elements from  $P$  having different (in terms of dual opposing) values (the proof in Pliszka and Unold (2011)).

As a conclusion of the above Theorem 1 we have convenient

**Theorem 2.** If a primary population  $P \subseteq H^n$  contains the pair of polar chromosomes, then the whole space  $H^n$  is a ancestral population, where two points  $h_t$  and  $h_k$  in  $H^n$  are called polar chromosomes if and only if for each coordinate these points have opposite values.

For example, having two polar chromosomes  $h_0$  and  $h_7$

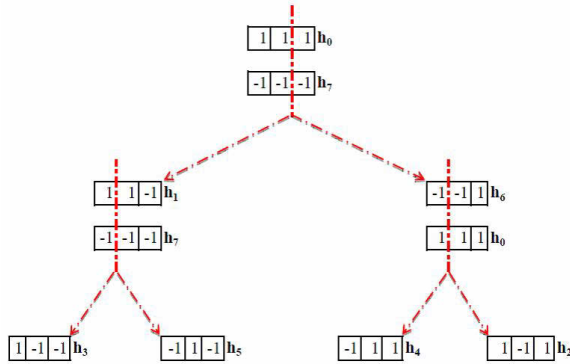


Figure 1: Exemplary ancestral binary population in Hadamard encoding with the primary population  $h_0$  and  $h_7$ .

as a primary population from  $H^3$  we are able after three crossovers obtain all 8 chromosomes from the space in question, according to Theorem 2 (see Figure 1). What is interesting is that more crossover operations are needed when using natural selection and random points of crossing.

Theorem 2 allows us to determine the number of necessary and sufficient one-point crossovers, we need to recover the entire space  $H^n$  from the two polar chromosomes.

**Theorem 3.** Any algorithm established to restore the entire space  $H^n$  from two polar chromosomes with a one-point crossover operator needs at least  $2^{n-1} - 1$  operations (the proof is omitted).

Moreover, it is possible to construct such an algorithm, which reconstructs the whole binary space exactly in  $2^{n-1} - 1$  steps.

Note that introduced representation allow us to distinct and classify different populations, what is more to penetrate into the potential future directions of their evolution regardless of the selected crossover algorithms, selection of parents, or the elimination of individuals. Having Theorem 3 we are able to compare GAs in terms of efficiency and optimization.

## References

- Hu, T. and Banzhaf, W. (2010). Evolvability and speed of evolutionary algorithms in light of recent developments in biology. *Journal of Artificial Evolution and Applications*, doi:10.1155/2010/568375.
- Novak, M. (2006). *Evolutionary Dynamics*. Harvard University Press, Cambridge, Mass, USA.
- Pliszka, Z. and Unold, O. (2011). How to predict future in a world of antibody-antigen chromosomes. In Ganzha, M., Maciaszek, L., and Paprzycki, M., editors, *Proceedings of the Federated Conference on Computer Science and Information Systems*, pages 91–96. IEEE.

Pliszka, Z. and Unold, O. (2012). On some properties of binary

# The only wrong cell is the dead one: On the enactive approach to normativity

Manuel Heras-Escribano<sup>1</sup>, Jason Noble<sup>2</sup> and Manuel de Pinedo<sup>1</sup>

<sup>1</sup> Departamento de Filosofía I, Universidad de Granada, Spain

<sup>2</sup> School of Electronics and Computer Sciences, University of Southampton, UK  
herasescribano@gmail.com

## Abstract

In this paper we challenge the notion of ‘normativity’ used by some enactive approaches to cognition. We define some varieties of enactivism and their assumptions and make explicit the reasoning behind the co-emergence of individuality and normativity. Then we argue that appealing to dispositions for explaining some living processes can be more illuminating than claiming that all such processes are normative. For this purpose, we will present some considerations, inspired by Wittgenstein, regarding norm-establishing and norm-following and show that attributions of normativity to non-social agents are deeply paradoxical. The main conclusions of our discussion are: (1) circular and internal explanations centred on the stability of living systems are insufficient to account for processes where the environment plays an important role, such as adaptation. Enactivism is not an explanatory alternative to evolutionary biology but needs it as a complement to accounts focused on the internal self-assembly of organisms; (2) though we share enactivism’s anti-representational spirit, we argue that ecological psychology can offer a better account of perception.

## Enactivism’s natural norms

Enactivism is often presented as the new paradigm for explaining cognition (Stewart et al., 2010). It is based on the assumption that cognition, rather than being a matter of abstract calculus and manipulation of internal representations in the head, is a spatio-temporally extended and dynamical process in which an embodied agent is meaningfully dealing with its environment in order to adapt itself to it. But this is not enough for defining enactivism. In fact, all these assumptions and tools have already been endorsed and developed by other anti-cognitivist theories, such as ecological psychology (Gibson, 1966, 1979). So, what’s new about enactivism? For some enactivists (Di Paolo, 2009) their theory provides a definition of agency in which the new embodied, extended and anti-representational cognitive science can rely on. Enactivism was born as a biological theory that emphasized the continuity between life and mind (Canguilhem, 1965; Maturana and Varela, 1987). Among the impressive achievements of the theory, perhaps the most important was the change of focus in thinking about living creatures: these are not seen as mere compounds of parts selected by evolution, but as whole agents individuated from their environment in terms of their internal structure. This

structure or system is based on different networked processes (such as metabolism and the different processes of the nervous system, for example) and it is taken as a unity. This is to say that the system as a whole provides stability, and the processes of this system that result from its stable configuration are intended to keep this unity going. An agent, thus, is autonomous or self-sustained, and its goal is to keep this self-stability. This is the sense in which life is normative, according to enactivism. Although this idea was already being embraced by some philosophers (Canguilhem, 1965; Jonas, 1966) only the explicit analysis due to Maturana and Varela made it into a suitable starting point for thinking about cognition. Cognition is one species in the wider genus of adaptive processes. Adaptive processes come in the form of a coupling typically described in mathematical terms; something that has been called the ‘agent-environment coupling’. In the context of perceptual processes, which are a special kind of adaptation, this coupling is called the ‘sensorimotor loop’. A refined notion of adaptation (in the first, broader sense) was developed later in the enactivist framework for clarifying how all these concepts are interrelated (Di Paolo, 2005).

Given the former definitions, we can broadly distinguish between two varieties of enactivism. First, those that endorse the biological notion of agency as a self-sustaining system along with the idea that perception is based on a sensorimotor loop. Second, those that are only committed to this way of explaining perception and do not hold to the theory of biological agency that defines the first group. Among the latter authors we can find Noë (2005) and O’Regan (2012). Conversely, among the former theoreticians we can find Maturana and Varela (1987), Jonas (1966) and others. We will focus on the notion of ‘normativity’ provided by the latter group, and for this purpose we will analyze the most recent definition of this phenomenon: the one provided by Barandiaran et al. (2009).

This approach to normativity is given in terms of its co-emergence with individuality and action, and these three notions work as different conditions for agency. Also, this agency is at the service of the autonomy of the system. So, given the fact that living systems are autonomous or self-sustained, for enactive philosophers a prior assumption is required in order to understand how agency emerges. For the enactivists, the difference between machines and living beings is that for machines “no *intrinsic force or process* is lumping the components together, nor has the system as a whole (independently of us) a specific way of functioning and

*demarcating itself from the rest*” (Barandiaran et al., 2009; emphasis added). The individuality of the entity is not projected but recognized when we deal with living beings. The entity itself displays to us its own criterion of demarcation from the environment: an agent is then “a system capable of defining its own identity as an individual and thus distinguishing itself from its surroundings; in doing so, it defines an environment in which it carries out its actions” (Barandiaran et al., 2009, p. 3; the quotation appears in italics in the original).

But a question remains unanswered: what is the particular process (or force, as it is quoted above) that allows for this demarcation? The answer is nothing but its own actions: “agents define themselves as individuals as an ongoing endeavor and *through the actions they generate*” (Barandiaran et al., 2009, p. 3, emphasis added). So, agents, by means of their acting, demarcate themselves as independent entities — and in doing so, they define themselves and also define the environment in terms of exclusion.

Here we arrive at a crucial point: if the definition of individuality comes by means the agent’s actions, how do we differentiate mere random movements from genuine actions? When we talk of an action-perception loop in order to describe the interaction of the agent with the environment we cannot consider that both parts are equally active in the interaction. The coupling of a leaf flowing in the air establishes a symmetrical relation: the weight of the leaf and the force of the wind regulate the process with the same degree of implication. As we have seen, living beings are different: they act upon the environment and thus they can be demarcated from it. This is an asymmetrical relation. Agents and environments do not play the same role in the coupling. Hence the sensorimotor loop (or any other coupling) is not like the leaf-air coupling: it is something provoked by the agent. Actions are not random because the agent tries to achieve a certain goal with them. An action is a goal-directed, normative movement. That purposiveness demarcates actions from other movements. Thus, “*agents have goals or norms according to which they are acting*, providing a sort of reference condition, so that the interactive modulation is carried out in relation to this condition” (Barandiaran et al., 2009; emphasis added). The statement quoted above is highly revealing: goals and norms are used interchangeably (Barandiaran et al., 2009, p. 5, footnote 2), and these norms are the reference condition by which we can say that agents are acting. Furthermore, the coupling with the environment is carried out in relation to this normative character that specifies the kind of interactions with the environment that are defined as ‘actions’.

But what is a ‘norm’ from this enactive perspective? Is it a statement or an explicit rule like ‘the queen can move any number of vacant squares horizontally, diagonally or vertically’ or ‘do not feed the animals’? Clearly not. First, no linguistic competence is necessarily involved in the employment of this kind of norm, and nor is interaction with other agents required. Rather, it seems that some process is normative when it establishes and maintains the individuality or self-sustenance of the system: “self production is a process that defines a unity *and a norm*: to keep the unity going and distinct” (Di Paolo, 2005, p. 434; emphasis added). A process

that benefits adaptation is a “norm given by self construction” (Di Paolo, 2009, p. 50).

Now we have the whole picture of enactivist agency: agents are systems that individuate themselves from the environment by means of their actions, and those actions are described normatively. We can talk of a *co-emergence of individuality and normativity*: even though enactivists separate these as different conditions, they also explain in what sense the two concepts are co-extensive or interrelated. It can be useful to briefly return to the quotes cited above: “agents define themselves as individuals as an ongoing endeavor and *through the actions they generate*” and “*agents have goals or norms according to which they are acting*”. This amounts to saying that individuality is defined in terms of action and action is defined in terms of normativity. So, it is this “*deep circularity* and *entanglement* between networked processes, the self-maintaining conditions they generate and the interactions that the system establishes with the environment what [*sic*] makes agents so challenging to model and understand” (Barandiaran et al., 2009, p. 8; emphasis added). Recently, Barandiaran and Egbert (in press) modeled the normative behaviour of a unicellular agent based on these criteria. In their model they differentiated between derived and intrinsic normativity, and they claim that the latter is a central feature of living beings, which are able to establish and follow their own norms in order to keep up the self-sustainability of their structure and their ability to adapt to their environments. That is why, for these authors, enactivism is a new paradigm: because it establishes a theory of agency through which we can understand cognition and, specifically, the normative aspect of it.

## Dispositions and norms

The first set of examples that could clarify the notion of ‘normativity’ defined by enactivism comes from the philosophical discussion of dispositions. Several authors have previously appealed to dispositions in order to explain the behaviour of physical objects but also of biological or rational agents (Ryle, 1949; Molnar, 2004; Mumford and Anjum, 2011). We say that sugar has the disposition to dissolve when put into water, neurons have the disposition to open their sodium channels when they receive a stimuli, and humans have the disposition to laugh when they listen to a joke. For some authors, these dispositional properties are defined as intrinsic, first-order, and real properties of agents and objects (Molnar, 2009). By ‘first-order’ we mean that dispositions are properties instantiated in individuals. The claim is that these dispositional properties are intrinsic to their bearers because they do not depend on the existence of any other object. Given these features, we can say for example that the fragility of a piece of glass is a property instantiated in a particular item, and also that the existence of that property does not depend on the existence of any object other than the piece of glass. The realist commitment to the property comes with the conclusion that, given its individuality and intrinsicity, a disposition does not need to manifest itself in order to prove its existence. Glass can maintain the property of being fragile even when it never breaks. We do not need the continuous manifestation of a dispositional property in order to assume its existence. Based on this, a special feature of dispositions is their



directedness: dispositions are directed to their reciprocal dispositional partners rather than to their manifestations. These reciprocal partners are other elements of the same kind that, given the right circumstances, play the role of triggering the manifestation of the property. For example, imagine a sugar cube: a sugar cube has the property of being soluble even when it is not able to show its solubility (let's say, even when it is covered by plastic wrap while submerged in a glass of water). In this case, the disposition exists even when it is not manifested. Also, following Martin (2008), we can imagine that some chemical product A has the property of being soluble when mixed with another chemical product B even if product B does not exist in the universe (e.g., product B has never been synthesized because it would be too expensive to do so). In any event, our realist intuitions towards dispositions incline us to consider that the product A has the property of solubility even if it will never be manifested.

An interesting feature of dispositions is that, applied to biological agents, they cover abilities and natural reactions as well as learned and innate responses. We can say that a dog once had the disposition to growl when the master picked up its bowl, but now it has the disposition to sit down when the master does the same thing. Also, given all the features mentioned above, dispositions can be useful for explaining interaction with the environment in a non-representational way: from a dispositional perspective, the basic unit of analysis is not the agent, but the agent-environment coalition. Dispositions, thus, explain the expected behaviours of certain agents under specific conditions.

Given this account of dispositions, we think that these properties are very useful for explaining different processes and behaviours of living agents, which are highly context-dependent. Thus, we are going to provide an example by which we can differentiate between a dispositional state and a normative behaviour: this will help us to show how the difference between following a rule and manifesting a disposition is blurred in the enactive account of perception. We will conclude that the subsumption of the dispositional within the normative is not helpful for explaining the different cognitive states of living agents. Imagine this situation: Manolo is a heavy smoker. To say that someone is a smoker is to make a dispositional attribution. Manolo is a smoker even when he isn't smoking. Being a smoker is being disposed to smoke a cigarette in certain circumstances (being a heavy smoker is to be so disposed in most circumstances). For instance, Manolo has the disposition to light a cigarette every time he sees one, as if an internal force pushed him to do it. One day Manolo goes to the doctor and he is told that if he continues to smoke, he is very likely to develop a chronic respiratory disease. He realizes that he *must* stop smoking. This realization did not cancel his *disposition* to smoke, at least not in the short term. Nevertheless, it did stop him from *manifesting* the disposition. What was it that stopped a deep-seated disposition from manifesting itself? The answer is clear: a *norm*. A norm can inhibit the triggering of certain dispositions even when the circumstances are otherwise entirely suitable for the disposition to manifest itself. From this perspective, a norm can inhibit but it can also eliminate a disposition after a certain number of corrections. So, the first

difference between a disposition and a norm is that the first is intrinsic and internal, but the second is not. This is why the internal force persists even when the agent follows a rule imposed by another agent (such as the smoker and the doctor). Given these features we can understand now why a norm, then, is different from a disposition. It is clear when we look at examples involving humans, but can this notion of 'norm' be applied to neurons or bacteria? It seems that there is no room for this conceptual tool in the explanation of the behaviour of unicellular agents. *Who* can correct the cell for not behaving in a certain way? How can an intrinsic disposition of a bacterium be inhibited by a norm, if the only dispositions that we can find in bacteria are those that allow for its survival (or, at least, that allowed for the survival of its ancestors)? It seems that there is something in the context of bacteria that is missing if we want to apply the concept of 'norm' to them.

## Wittgensteinian norms

As we've seen, enactivists do not differentiate between *biological* dispositions and norms, and they label all these different processes 'normative'. This would be a minor problem if it were only a terminological issue. However, we think that the problem is also conceptual and ontological: what we have here is a disagreement regarding what a norm *is*, *how* the concept works and *in which context* it can be applied. We have seen that some enactivists claim that a solitary agent can both establish and follow its own norm. Now the question is whether that claim is acceptable. Is it intelligible to think of an agent who is able to establish and follow its own rules in isolation?

We think that Wittgenstein's discussion on rule-following is still very relevant to this question. In the well-known sections of his *Philosophical Investigations* devoted to this issue, he offers a battery of arguments to show that the answer should be negative (Wittgenstein, 1953, §§ 185-242). When we talk of following and establishing our own norm we are talking of establishing and following a special course of action. Wittgenstein wants us to imagine a situation in which somebody is teaching a pupil to count in a certain way and the teacher wonders why, after many repetitions, the student is still not doing it correctly. The first explanation of the student's behaviour is always to appeal to his natural reactions, to his natural inclinations, for answering one way rather than another. This suggests that we can distinguish between acting according to one's natural dispositions and acting correctly — acting according to a rule. So, following a rule seems to be something much more complex than naturally reacting. If all there is to following a rule was to act according to one's brute inclinations, then there would be no situation where learning could be thought to be necessary to coming to act *in the right way*.

If equating norm-following with acting according to one's unlearned natural dispositions is problematic, perhaps the enactivist, in her defense of the idea of biological norms, could appeal to the notion of interpretation. A sphere would be normative inasmuch as its inhabitants were capable of interpreting norms in such a way that their action was a case of following the rule under their interpretation. When discussing whether acting according to a norm can be

understood as offering an interpretation such that the action becomes subsumable by the rule, Wittgenstein comes back to the example of the pupil learning mathematics. After some successful exercises that seemed to show that he had mastered the use of the “+” sign (all involving numbers smaller than 1000), the teacher asks him “how much is  $1000 + 2$ ?” The student answers “1004”. When the teacher tells him that this is not the right answer, he defends himself claiming that he is doing exactly what he was told: “I did as before. Wasn’t the rule: add 2 up to 1000, 4 up to 2000, 6 up to 3000 and so on?” The student has managed to provide an interpretation of the rule behind the use of the “+” sign that covers all possible uses of the sign and is consistent with all of the examples he was exposed to during his learning. It is tempting to say that the pupil can act in accordance with his own criterion. At the very least, he seems to show a personal and systematic way to face stimuli after a number of repetitions and encounters with them. A defender of the idea of non-social, natural norms would argue that the habitual answer to the stimulus can become a norm (i.e., a well-established causal connection). In fact, on what else would the rule-following of an isolated agent depend than on its personal interpretation of the norm (i.e., on its own systematic way of reacting to a given stimuli)?

This way of understanding normativity seems deeply paradoxical: if acting according to a rule is no more than interpreting the rule in such a way that the action falls under it, then every action can be made out to accord with some interpretation of the rule and every action can also be made to conflict with an interpretation of the rule. Then there would be neither accord nor conflict here. If every idiosyncratic interpretation of the rule is right, then how we could say that somebody is wrong? It would seem that the concepts ‘right’ or ‘wrong’, which are tightly connected to the concept of ‘norm’, are of no use here. So, if everything is a norm, then nothing is a norm at all because nobody could distinguish what is normative from what is not. As Wittgenstein claims in § 201: “What this shows is that there is a way of grasping a rule which is not an interpretation, but which is exhibited in what we call ‘obeying the rule’ and ‘going against it’ in actual cases.”

Our claim in this paper is that the idea of a bacterium establishing and following norms is just as problematic as the idea that all there is to grasping a rule is to behave in a way that coheres with some possible interpretation of the rule. To talk about norms is to talk about the possibility of being right and wrong, and this in turns demands that the agent be capable of distinguishing between “it is correct” and “it seems correct to me”. Could anyone make such a distinction without having being corrected in the past? We believe not. Given the fact that the aspirant to being a rule-follower cannot be its own corrector, we claim that rule-establishing and rule-following need to be defined as a socially-mediated phenomenon. Norms can only emerge within a social context; norms are, then, social institutions. Norm-establishing is a social process. That is precisely why norms are external: because the criteria of correctness are shared across a community of agents.

The alternative, solipsistic conception of rule following makes following a rule analogous to speaking a private language (i.e., to following private, internal linguistic norms).

Norms must be guided by certain criteria that determine the correctness of their own applicability. These criteria are external in the sense that a single agent acting alone cannot establish them: if that were the case, senseless situations like the one discussed above would be common. But why is that situation ‘senseless’? Because if somebody follows a rule and she cannot distinguish between following it and not following it, she cannot guarantee that she is following the rule in the right way. This point is explicitly stated by Wittgenstein: “Hence it is not possible to obey a rule ‘privately’: otherwise thinking one was obeying a rule would be the same thing as obeying it” (Wittgenstein, 1953, § 202). So, where do rules or norms come from? Don’t they come from the agents that establish them? Sure, but this is not the same as saying that a solitary agent could be involved in the process of establishing a rule. Norms can only emerge within a social context.

Rule-establishing cannot be a private exercise, but what about norm-following? The enactivist claims: “even if the origin of some norms does not fully lie within the individual (e.g., social norms) it is always the individual who internalizes them” (Barandiaran et al., 2009, p. 6). What sort of process could this internalization be? For the purpose of answering this question, let’s rescue another classic example from Wittgenstein, that of the beetle in the box (Wittgenstein, 1953, § 293). Let’s assume that everyone in their own case knows how to follow a rule because they have internalized it. Each of us would walk around carrying a box and calling what is inside ‘a beetle’ — or, better, ‘a norm’. Nobody can see inside anyone else’s box, and everyone knows what a beetle (or a norm) is only through looking inside their own boxes. On the other hand, we all know how to use the concept ‘beetle’ or ‘norm’. Suppose that in fact we all have different things in our boxes (or even imagine that there is nothing at all in them). The key point here is that the object in the box plays no role at all in our understanding of how to use the concept. In the same vein, we do not need to look inside us or appeal to any inner state to know what following a rule is. The criteria are *outside* the individual; they are located in the social community. They are *shared*. But they are not *objects*. We do not need to look for them as if they were part of our internal machinery. This is why norms are not individual-internal, but social-external processes, both when they are established and when they are followed. Being goal-oriented and having conditions of success and failure is necessary but not sufficient for being normative. Normativity also demands awareness of the possibility of error, training, habit and social learning.

## Conclusions and further work

In this paper we have offered three inter-related arguments against enactivism’s insistence on talking about norms at the level of simple, non-social agents: (1) the co-emergence of individuality and normativity is just taken for granted because the claim that these are mutually supporting ideas is viciously circular. A robust notion of agency related to an evolutionary history of adaptation and selection is sufficient to account for the singularity of living systems. (2) A notion of normativity as vague as the one offered by the enactive theory blurs the distinction between dispositional, individual, intrinsic natural processes and the social, external and institutional ones that

can inhibit the former group. A clear separation between phenomena defined by mere conditions of success and failure and phenomena characterized by correctness conditions makes explicit such a distinction. (3) The concept 'norm' can only be applied to what enactivists call 'social norms': the Wittgensteinian discussion of rule-following shows that there is something deeply paradoxical in thinking of the behaviour of an agent considered in isolation as being governed by norms. As usual, we should not be led to confusion by etymology: full-blown normativity and self-regulated behaviour are to be distinguished. A single agent cannot establish a rule because it is acting according to its dispositions; acting according to one's dispositions and acting according to a rule are not the same thing; and, finally, there is no need to appeal to any internalization of the norm to explain how agents follow them.

The behaviour of a cell is manifestly suitable for explanation in dispositional terms, because it cannot be divorced from its environment. We can say that its behaviour is rich enough to qualify it as 'goal-directed'. But it is not normative because there is no socially-established norm that could inhibit any of the cell's intrinsic dispositions. Inasmuch as the criteria of correctness of that hypothetical norm are not shared, the cell could not possibly distinguish between instances when it is acting according to the norm and instances when it is not. Neither could we: the distinction between failure due to the cell's behaviour and failure due to, say, a hostile environment cannot be made for actions that are mere manifestations of dispositions. Our claim is that a cell's behaviour may be insufficient to guarantee its survival in some environment, but that such failure does not entitle us to consider the behaviour incorrect. This is why the only wrong cell is the dead one.

Our qualms with enactivism's excessively liberal use of normative considerations is no obstacle to our sympathy with enactivism's anti-representationalist commitments, as well as its emphasis on embodiment, situatedness, the active character of perception, and the centrality of the agent as a whole. We also agree that perceptual relations with the environment can be explained by means of looping processes. However, we think that this anti-representational approach to cognition is better developed by ecological psychology (Gibson, 1979). Gibsonians gave an account of perception in a way that is much more externalist, biosemiotic, and structure-independent than the sensorimotor contingencies defended by the enactive view. This is the reason why ecological psychology has provided a better account of learning than enactivism, even though they start with the same anti-representational assumptions (Jacobs and Michaels, 2007). Enactivism is too closely focused on the internal structure of the organism and concedes too little attention to the explanatory role that the environment plays with respect to perception and action. We also depart from the enactivists regarding their faith in autopoiesis being the best explanation of every aspect of biological processes: from the emergence of agency to the emergence of perception. We also do not see autopoiesis as being the best explanatory framework for processes such as adaptation and cognition. In fact, as we have seen, the enactive, co-emergent explanation consists in subsuming all biological processes into just one: the recursive loop made by all systems of every organism. This may well be the best

answer to the question of how all organisms are able to maintain their stability through time, but that does not amount to defining agency, adaptation, cognition and the rest of the set of biological processes at once. Take the example of adaptation, a process the enactivists sometimes call 'adaptivity' (Maturana and Varela, 1984; Di Paolo, 2005). This process is based on the recursive loop we have mentioned, and we can apply the same logic of recursivity to the relations of the agent with the environment: an adaptive capacity is one that is able to regulate its relation with the environment in order to keep the agent within a state of viability. Organisms can detect tendencies in which the agent approaches (or recedes from) the boundary of viability. As any biologist would concede, this formulation is insufficient to account for adaptation in the full sense. Adaptation is a trait that contributes to the fitness and survival of individuals but it also needs to be explained as the result of processes of natural selection, and reference eventually needs to be made to species and populations (Darwin, 1859; Huxley, 1942; Williams, 1966; Mayr, 1983). If we want to give a full account of why an agent is adapted, we necessarily need to appeal to its evolutionary history and talk about how natural selection works. This is a question answered only at the macroscopic level and by means of reverse engineering (Dennett, 1995), not by looking at the looping processes of individual agents. We think that the excessive emphasis on the logic of looping processes is leading enactivism towards an underestimation of natural selection, the role of populations, and the different levels of explanation involved. Not all questions in biology are answered by redirecting the answer to the looping processes of self-sustenance of individual agents. Some questions (why a trait has evolved this way rather than that way, *why* we have perceptual system at all, etc.) are answered by appealing to the supra-agential realm and this means by appealing to how natural selection works. Whereas other questions (*how* do we perceive, etc.) are answered appealing to looping processes (ecological, perception-action loops). There are different questions addressed by different levels of explanation. We do not think, as enactive theorists seem to endorse, that all biological processes can be explained by means of their looping and co-emergent logic and by appealing to the autopoiesis of individuals.

We think that this philosophical discussion is clearly of interest for computer scientists for two very important reasons: it is important not to confuse levels of analysis and also not to misattribute properties or predicates to agents that do not fulfill the right criteria of application. An unicellular agent cannot be wrong because there is no room for norms in its behaviour. For following a norm some conditions are needed: (1) a community, (2) the possibility to err, (3) correction criteria for the right application of a concept or a right way of behaving in a certain context, (4) the possibility to differentiate between following a norm and thinking that one is following a norm, (5) to be sanctioned by a community in order to understand which are these criteria and how to differentiate between what one thought she was doing when following a rule and what she was really doing. Even when a Wittgensteinian strategy is not committed to offer necessary and sufficient conditions for defining a concept, these previous points can summarize more or less some features that are common to any notion of 'norm'. We think that,

summarized in the previous points, a computer scientist does not offer anything new when he claims that a unicellular isolated agent can follow norms. That claim only shows that, even when he could design a really good model, the scientist never got the conceptual point of what a norm is. Even though it is surely possible to describe all of the different levels of agency from a naturalistic viewpoint (making use of our best empirical evidence from the biological sciences), introducing the most complex concepts, such as ‘normativity’, when studying the most primitive forms of agency is not a good strategy. A better strategy would be to focus on what have been called the ‘major transitions in evolution’ (Maynard Smith and Szathmáry, 1995); that is, to focus on the conditions under which new organizational levels appear rather than taking them for granted. Unlike enactivism, we reject the idea that the naturalization of normativity can be made by normativizing nature: all that is rational is real, but not all that is real is rational.

## Acknowledgements

All three authors of this paper are funded by the Spanish Ministerio de Ciencia e Innovación under the research project “Dispositions, Holism and Agency” (FFI2010-19455).

## References

- Barandiaran, X. E. and Egbert, M. D. (in press). Norm-establishing and norm-following in autonomous agency. To appear in *Artificial Life*.
- Barandiaran, X.E. Di Paolo, E. and Rohde, M. (2009). Defining agency. Individuality, normativity, asymmetry and spatio-temporality in action. *Journal of Adaptive Behavior*, 17 (4) 367-386 in Rohde, M. and Ikegami, T., editors, special issue on Agency in Natural and Artificial Systems.
- Canguilhem, G. (1965). *La connaissance de la vie*. Vrin, Paris.
- Darwin, C. (1859). *On the Origin of Species*. John Murray, London.
- Dennett, D. C. (1995.) *Darwin's Dangerous Idea*. Simon and Schuster, New York.
- Di Paolo, E. (2009). Overcoming autopoiesis: a enactive detour on the way from life to society. In Magalhaes, R. and Sanchez, R., editors, *Autopoiesis in organization and information systems*, pages 43-68. Elsevier.
- Di Paolo, E. (2005). Autopoiesis, adaptivity, teleology, agency. *Phenomenology and the Cognitive Sciences*, 4: 429-452.
- Gibson, J. J. (1966). *The Senses Considered as Perceptual Systems*. Houghton-Mifflin, Boston, MA.
- Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Houghton-Mifflin, Boston, MA.
- Huxley, J. S. (1942). *Evolution: the Modern Synthesis*. Allen and Unwin, London.
- Jacobs, D. and Michaels, C. (2007). Direct Learning. *Ecological Psychology*, 19: 321-349.
- Jonas, H. (1968). Biological foundations of individuality. *International Philosophical Quarterly*, 8 (2): 231-251.
- Martin, C. B. (2008). *Mind in Nature*. Oxford University Press, Oxford.
- Maturana, H. and Varela, F. (1987). *The Tree of Knowledge*. Shambhala, Boston, MA.
- Maynard Smith, J. and Szathmáry, E. (1995). *The Major Transitions in Evolution*. Oxford University Press, Oxford.
- Mayr, E. (1983). How to carry out the adaptationist program? *The American Naturalist*, 121 (3): 324-334.
- Mumford, S. and Anjum, R. (2011). *Getting Causes from Powers*. Oxford University Press, Oxford.
- Molnar, G. (2004). *Powers: a Study in Metaphysics*. Oxford University Press, Oxford.
- Noë, A. (2005). *Action in Perception*. The MIT Press, Cambridge, MA.
- O'Regan, J. K. (2012). How to build a robot that is conscious and feels. *Minds and Machines*, 22 (2): 127-136.
- Ryle, G. (1949). *The Concept of Mind*. Hutchinson, London.
- Stewart, J. Gapenne, O. and Di Paolo, E. editors (2010). *Enaction: Toward a New Paradigm for Cognitive Science*. The MIT Press, Cambridge, MA.
- Williams, G. C. (1966). *Adaptation and Natural Selection*. Princeton University Press, Princeton, NJ.
- Wittgenstein, L. (1953/2001). *Philosophical Investigations*. Blackwell, Oxford.



# Controlling Task Distribution in MONEE

Evert Haasdijk<sup>1</sup> and Nicolas Bredeche<sup>2</sup>

<sup>1</sup>Vrije Universiteit Amsterdam, <sup>2</sup>UPMC Univ Paris 06, UMR 7222, ISIR, F-75005, Paris, France  
e.haasdijk@vu.nl, nicolas.bredeche@isir.upmc.fr

## Abstract

The MONEE framework endows collective adaptive robotic systems with the ability to combine environment- and task-driven selection pressures: it enables distributed online algorithms for learning behaviours that ensure both survival and accomplishment of user-defined tasks. This paper explores the trade-off that must be reached between these two (possibly contradictory) requirements, in the case where a foraging task is defined by the user. In particular, we study the impact of enforcing specialisation (i.e. the collective must acquire two mutually exclusive foraging skills) as well as the mechanism for tuning the level of specialisation in an on-line fashion. Results show that the actual behaviour of the collective system can be guided on request during the course of evolution in order to achieve a particular distribution of specialisations, albeit within a certain range of values.

## Introduction

The work in this paper is inspired by a vision of a collection of robots that evolve to survive and operate in an environment where human control can be effected only intermittently. In such circumstances, the robots have to act autonomously, without direct human intervention. They must therefore survive long periods without any guidance and when they do receive guidance, it is at a considerable delay. The environment is not completely known at deployment time and it changes over time, as do the tasks that the robots have to complete. Therefore, the robots must adapt to survive the environment and to perform their tasks.

The environment in which robots operate indirectly circumscribes goals for the population of organisms to survive and evolve, but does so without specifying objective functions: the robots must for instance move about to spread their genomes, or they must maintain their energy levels, but these goals are not defined directly: it is just that robots that display this behaviour get more opportunities to procreate. By virtue of its similarly unbounded nature, biological evolution has resulted in the high levels of adaptability and robustness that we see in natural living organisms. To exploit this creative potential in a system of evolving robots (or robot controllers), we would want to give evolution

as much freedom as possible, pushing for open-ended, unbounded adaptivity, unconstrained by user-defined objective functions.

On the other hand, if the system is to be of any practical relevance, the robots must of course also perform user-defined tasks, pushing for specific, crisply defined task-related objectives.

Evolution has been employed to achieve both of these facets. Artificial Life research abounds with examples of objective-free evolutionary systems since the 1908s (Langton, 1989, 1995). In such experiments, evolution serves as a force for adaptation. Evolutionary robotics research typically employs evolution as a force for optimisation when it focusses on the task-driven aspect (Nolfi and Floreano, 2000).

Balancing these two aspects of evolution –environment-driven adaptation and task-driven optimisation– represents a vital step towards implementing our vision of autonomous, functional, responsive and self-sufficient robot collectives.

In earlier work, we presented the MONEE (Multi-Objective aNd open-Ended Evolution) to solve the problem of combining objective-free and task-driven evolution in a single algorithmic framework (Haasdijk et al., 2013).

The principal idea behind MONEE is to employ concurrently two selection mechanisms in different roles: environmental selection for open-ended evolution and parent (or mate) selection for task-driven adaptation. As the 'Multi-Objective' part of the name implies, MONEE accommodates settings with multiple tasks. Jones and Mataric noted that collectively tackling multiple tasks also entails a division of work (2003). If there are multiple tasks, the population of robots as a whole must tackle all of them, even though individual robots may specialise in only a subset. To cope with such cases the MONEE framework uses a market mechanism. This mechanism regulates task-based rewards during mate selection according to the market logic that scarcity increases worth. In our multiple task context this implies that tasks that only a few robots (can) perform yield relatively high rewards and therefore higher selection probabilities.

We showed that the MONEE paradigm does indeed allow

the robots to adapt their behaviour to the environment as well as to multiple tasks. Also, MONEE's market mechanism is crucial to keep the population from focussing exclusively on easier tasks, even when the environment induces specialisation in particular tasks at the individual robot level (Haasdijk et al., 2013).

The market mechanism offers an intriguing possibility for intervention in the adaptive process: users can define premiums for particular tasks to (de-)emphasise their importance and promote or prevent their take-up by the robots. This amounts to defining an exchange rate between credits earned for the various tasks. Such premiums provide a straightforward and intuitive method for human-*on*-the-loop intervention in the behaviour of the robot collective.

We perform an experimental analysis of the influence of premium settings in an implementation of the MONEE paradigm where a simulated population of robots has two tasks: it must collect red and green pucks. The experiment is set up so that controllers for each task must be learned separately. In particular, our research questions are:

- To what extent can a premium direct the focus of the robot swarm to a particular task?
- Does a negative premium prevent the robots from displaying particular behaviour?
- How does swarm behaviour react to changing premium settings?

### Related Work

Bredeche et al. describe meDEA (Bredeche et al., 2012), an open-ended evolutionary algorithm where autonomous robots move around an arena while continually broadcasting their genome over a short range. Meanwhile, they also receive genomes from other robots that come in communication range. When a robot's lifetime expires, it randomly selects one of the received genomes, modifies that using mutation and starts a new life of broadcasting this new genome. This set-up promotes, with only environmental selection, robot movement through the environment: genomes that cause the robot to move around a lot are spread at a much higher rate than genomes that cause their host to stand still.

Similar settings have been extended with forms of parental investment, for instance in Mascaro et al. (2005); Ventrella (2005); Schwarzer et al. (2010). In artificial life parental investment is often used to give the offspring a starting value of (virtual) energy (Menczer and Belew, 1996; Menczer et al., 1994; Burtsev et al., 2001; Scheutz and Schermerhorn, 2005) and a parent's energy level is often linked to task performance (e.g., agents tasked with eating grass to gather energy in Burtsev et al. (2001)). Distributed on-line evolutionary systems such as Watson et al.'s embodied evolution similarly employ task-related (virtual) energy to determine parent and survivor selection (Watson et al.,

2002; Wischmann et al., 2007), typically considering single tasks. These experiments showed that task-related virtual energy (equivalent to credits for appropriate behaviour) is an effective way to guide evolutionary adaptation to tackle tasks.

Market-based schemes provide a well known solution to the task allocation problem in multi-agent and multi-robot settings, for instance in (Walsh and Wellman, 1998; Tang and Parker, 2007).

Fitness sharing is a well-known technique that was introduced to promote genetic diversity and so prevent premature convergence in evolutionary algorithms. With fitness sharing, an individual's fitness is reduced if there are many similar (in terms of their genetic makeup) individuals in the population. Traditionally, fitness sharing is not necessarily associated with multiple objectives, but with maintaining diversity in general – typically, but not exclusively, in single-objective settings.

### MONEE: Multi-Objective & Open-Ended Evolution

As mentioned above, earlier work showed that MONEE effectively combines environment- and task-driven adaptation (Haasdijk et al., 2013). The population of robots shows similar adaptation to the environment with MONEE as it does with its purely environment-driven counterpart meDEA. In addition, the robots learn to perform puck-collecting tasks. They equitably distribute the collective foraging effort over different puck types, even when one type is more prevalent than the other or when the environment inhibits individual robots gathering multiple types of puck.

The robot –actually, their controllers'– lifecycle in MONEE consists of two phases: life and rebirth. The robot controllers have a limited, fixed, lifetime during which they perform their actions; moving about, foraging, et cetera. When their lifetime ends, they enter a rebirth phase and become 'eggs': stationary receptacles for genomes that are transmitted by passing live robots. This rebirth phase also lasts a fixed amount of time, and once this has passed, the egg selects parents from the received genomes to create a new controller. The robot then reverts to the 'life' role with this new controller. Thus, robot controllers can procreate by transmitting their genome to eggs, and the more eggs a robot inseminates, the more chances it has for procreation. Because the transmission of genomes is continuous and at close range (e.g. through infrared), the more a robot moves about the arena, the better its chances of producing offspring. This aspect of MONEE is open-ended in the sense that it is objective-free: there is no calculated performance measure that defines the chances of being selected as parent, there is no task. Only the environment and robot behaviour dictates what robots may or may not become parents.

To add task-driven parent selection to this basic evolutionary process, the robots can, during their lifetime, amass

credits by performing tasks. For instance, a robot could get one credit for every piece of ore it collects, one for successfully solving some puzzle, and so on. If multiple tasks are defined, the robots maintain separate counts for the credits awarded for each task, for instance one counter for the pieces of ore collected and another one for the number of puzzles solved. When a robot inseminates an egg, it passes the current credit counts along with the genome and the egg uses that information to select parents when it revives. This scheme is reminiscent of parental investment, but it differs subtly yet crucially from most parental investment schemes: a parent does not actually invest when impregnating an egg because the credits aren't *transferred* but *copied* at no cost to the parent.

When a robot's egg phase finishes, it compares the parents' credits for each genome it has received. To enable this comparison across tasks, the egg calculates an exchange rate between tasks. This ensures that genomes that invest in tasks for which few credits are found overall (presumably hard tasks) are not eclipsed by genomes that favour easier tasks.

The credits relate task performance to reproductive success: besides the open-ended goal of 'merely' transmitting genomes to eggs, robots must also become proficient at the defined tasks for these genomes to be selected. The more proficient a robot is at a task, the higher its chances of procreating. The comparison of credits across multiple tasks introduces an exchange rate between the earnings per task: the more common credits are for a particular task, the less their worth and vice versa. Thus, parent selection becomes a marketplace for skills and features that the user requires. This system naturally caters for multi-objective approaches.

MONEE's market mechanism is similar to fitness sharing in the sense that it also reappraises fitness, favouring tasks that are less commonly tackled by robots in the population. A crucial difference with traditional fitness sharing is that MONEE considers an individual's *behaviour*, not its genetic make-up (reminiscent of *syntactic* fitness sharing in genetic programming (Nguyen et al., 2012)). Hence, it does not promote genetic, but behavioural diversity: it modifies fitness not to prevent premature convergence, but to ensure that the robot population tackles multiple tasks.

It also allows the user to prioritise tasks in a straightforward manner: the user can influence the credit comparison by defining a premium for some or all of the tasks. For instance, if she deems collecting ore more important than solving puzzles, she can define a premium for collecting ore; the credits earned through this task are then multiplied by the premium. Compared to not defining a premium (or defining a premium of 1), setting a premium  $> 1$  increases the payoff for the relevant task, setting it between 0 and 1 reduces it, while setting it to a negative value causes the robot adaptation to shy away from the task.

The pseudo-code in algorithm 1 details the credit comparison market mechanism with premiums defined.

```

for every defined task do                                // total credits
  for every received genome do
    creditstask ←
    creditstask + (premiumtask · genome.creditstask)
  end
  creditsoverall ← creditsoverall + creditstask
end
for every defined task do                                // exchange rate per task
  ratetask ←  $\frac{\text{credits}_{\text{overall}}}{\text{credits}_{\text{task}}}$ 
end
for every received genome do                              // credits per genome
  for every defined task do
    genome.rating ← genome.rating + (premiumtask ·
    genome.creditstask · ratetask)
  end
end
                                                                    // select, mutate and revive
parent ← rank_based_selection(received_genomes)
child ← mutate(parent)
reactivate(child)

```

Algorithm 1: MONEE's market mechanism

## Experimental Set-up

We implemented the MONEE algorithm in a simple 2D simulator called RoboRobo (Bredeche et al., 2013). In our experiments, 100 simulated e-pucks are placed in an environment that contains obstacles and pucks. The sides of the square arena are roughly 330 robot body lengths long (1024 pixels in the simulator), and it contains a number of obstacles (see Fig. 1). We run 64 repeats of each experiment.

There are two types of puck: green and red, defining a concurrent foraging scenario. Concurrent foraging is a variation of regular foraging where the arena is populated by multiple types of objects to be collected (Jones and Mataric, 2003), rather than just a single resource. In our case, these objects are green and red pucks and the collection of each different colour is a different task. The pucks are distributed throughout the arena, and they are immediately replaced in a random location when picked up. The (re-)placement of pucks is governed by a 2D gaussian distribution centred on the middle of the arena and with  $\sigma$  of half the arena width. The robots move around the arena, spreading their genome as they encounter eggs and dying when their allotted time has passed. They collect pucks simply by driving over them and the more pucks they gather, the more likely their genome is to be selected once an egg they impregnated revives.

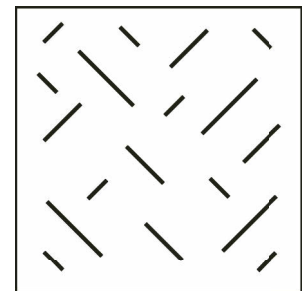


Figure 1: Experiment arena.

To detect pucks, the robots have 16 sensors that detect either red or green pucks (i.e., 8 sensors per puck-type). Each set of 8 sensors is laid out in the same manner as the stan-

ward e-puck infrared sensors: 6 face forward, 2 face to the rear. Because individual puck sensors only detect a single type of puck, collecting one type of puck is a task distinct from (but very similar to) collecting the other type of puck. Thus, behaviour to collect either type of puck has to evolve separately.

Each robot is controlled by a single-layer feed forward neural network which controls its left and right wheels. The inputs for the neural network are the robot's puck and obstacle sensors.

The robot's genome directly encodes the neural network's weights (3 types of sensor  $\times$  8 sensors  $\times$  2 outputs plus 2 bias connections plus 4 feedback (current speed and current rotation to either output) = 54 weights) as an array of reals.

As mentioned, the robots alternate between periods of explorative puck gathering and motionless genome reception. To prevent synchronised cycles among the robots, we add a small random number to each robot's fixed lifetime. This forces desynchronised switching between life and rebirth even though our runs start with all robots perfectly in sync at the first time-step of their lifetime.

At the end of the egg phase offspring is created by selecting a parent from the received genomes as shown in algorithm 1 and mutating the weights in that genome using gaussian perturbation with a single, fixed mutation step size  $\sigma = 1$ . This single-parent, mutation-only scheme is common in evolution strategies that are known to perform well on problems with continuous-valued genomes (Beyer and Schwefel, 2002).

Note that MONEE does not prescribe any particular controller implementation nor any choice of variation operator. The implementation we chose here of an artificial neural network with the weights encoded as real-valued genes provide a convenient, flexible and well-established representation.

Table 1 summarises the experimental set-up. The paragraphs below describe our experiment's variants in detail. Code for the experiments is available at [http://pages.isir.upmc.fr/evorob\\_db/moin.wsgi](http://pages.isir.upmc.fr/evorob_db/moin.wsgi).

**Premium** It is straightforward to (de-)emphasise particular tasks in MONEE by simply putting a premium on credits earned for that task. To investigate how premiums influence adaptation, we apply premiums ranging from -1 to 100 to the task of collecting green pucks, including a number of runs where the premium is redefined during the run. The premium for red pucks remains constant at 1.0.

During parent selection, the premium is used as a multiplication factor for the number of green pucks collected. Thus, with a premium set to -1, robots collecting green pucks are penalised. A premium of 0 means that there is no benefit to collecting green pucks: only red pucks are considered for parent selection. A premium of 1 means that red and green pucks contribute equally to the chance of a

Experiment details	
Robot group size	100
Simulation length	1,000,000 time-steps
Number of repeats	64
Number of pucks	500, 150 or 50 green, 500 or 150 red
Arena	See fig. 1
Premium settings	-1,0,1,2,5,10,20,50,100
Controller details	
Controller	Perceptron neural net
Input nodes	8 obstacle sensors, 16 puck detectors, 2 bias and 2 recurrent nodes
Output nodes	2 (left and right motor values)
Evolution details	
Representation	Real valued vectors
Chromosome length	54
Mutation	Gaussian $N(0, 1)$
Parent selection	Rank-based
Robot lifetime	2000 time-steps
Egg-phase	200 time-steps
Comm. range	ca. 9 body lengths

Table 1: Experimental set-up

genome being selected, and higher values increase the importance of collecting green pucks.

**Mutually exclusive skills** Equitable task distribution is more challenging when the tasks that the robots must perform are to some extent exclusive, for instance because they require irreconcilable skills. To test how premium settings affect the MONEE paradigm in such situations, we also run experiments where the environment constrains multi-skilled robots so that the robots must specialise in collecting one type of puck. Without this constraint, robots can collect green and red pucks equally well without any penalty when selecting both or merely one colour. In the mono-skill experiments the speed of robots depends on their specialisation level: the robot's speed is multiplied by the ratio of most prevalent pucks it has collected. Thus, if a robot collects exclusively pucks of one colour, its speed is maximal. If it collects 75% green (or red) pucks, its speed is reduced by 25% and if it collects red and green pucks in equal amount, the speed is halved. This penalty is recalculated whenever a robot picks up a puck. It is important to note that this is enforced by the environment, not during the parent selection phase when an egg revives. The environment causes specialising robots to move faster, so that they perform better than non-specialised robots: their higher speed allows them



to collect more pucks during their lifetime, but more importantly, it allows them to impregnate more eggs. This results in an increase in the proliferation of mono-skilled genomes without altering the selection process inside the eggs.

**Distribution of pucks** Another determinant for the difficulty of task distribution in our experiments is the ratio of puck colours. This can be seen as a proxy for having a difficult (rare pucks) and an easy (common pucks) task. To determine the impact of setting premiums with an uneven distributions of pucks, we run two variants of our experiments: one with 150 pucks of each colour, one with 50 green and 150 red pucks. We perform additional runs with denser spreads of pucks where there are 500 pucks of each colour.

**Changing premium** A last set of experiments explores the evolutionary dynamics in the context of changing premiums. The rationale is the following: what would be the effect on the ratio of harvested puck colours if the premium is reset on-the-fly by a human supervisor? Then, what happens if the premium is changed back to its initial value after awhile? The system dynamics would be more predictable if the harvested pucks ratio matches the original figures, that is evolutionary dynamics always converge to the same ratio values, independent from the initial conditions. It may, however, be expected that the evolutionary dynamics are affected by the behaviour from where it already converged (i.e. the ratio depends from the actual premium *and* from where evolution starts). To explore the influence of changing premiums on-the-fly we use the following set-up: in a setting with the same number of red and green pucks (150 of each) the premium is initially set to 10. After 500,000 time steps, the premium is changed to 1 for 250,000 time steps. It is then reset to 10 for the remainder of the experiment.

## Results and Analysis

**The Effect of Premiums** Figure 2 shows the mean total number of pucks collected in the experiments with 150 green and red pucks. Setting a negative premium predictably decreases the total number of pucks collected: the robots learn to avoid green pucks, in effect halving the number of available pucks. Setting the premium to 0 in the mono-skilled (i.e. with specialisation) environment still results in much lower levels of collected pucks because, again, the robots learn to keep away from green pucks and so avoid the environment's speed penalty for generalist behaviour. This penalty does not apply in the multi-skilled environment (i.e. without specialisation), and the number of pucks collected for premium 0 is markedly higher than with premium -1. The robots now pick up green pucks accidentally and they can take more direct paths to red pucks because they do not have to avoid green pucks. Setting the premium to 1 increases the number of pucks collected: robots now actively

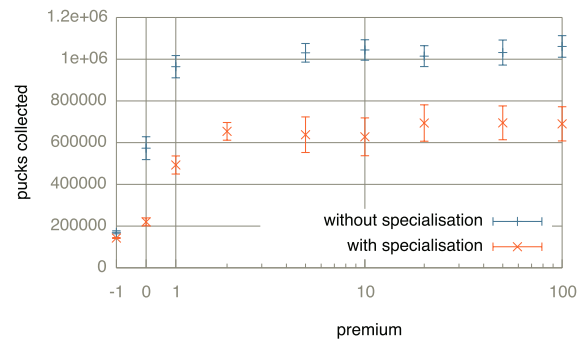


Figure 2: Mean total number of pucks collected over the whole run for different premium settings with 150 pucks of each colour. The vertical bars indicate the 99% confidence interval (over 64 repeats).

seek both types of puck. The mono-skilled environment still causes individual robots to avoid one type of puck or the other, therefore the number of pucks is lower than in a multi-skilled setting. Higher premium values slightly increase the number of pucks collected, but among these values it does not change appreciably.

To assess the impact of premium settings on the task distribution among the robot collective we consider the ratio of green pucks collected ('green puck ratio') over all collected pucks. Figure 3 shows how this ratio develops over time for different premium settings in the experiment with 150 pucks of each colour. Initially, the robot collective always gathers green and red pucks in a 50-50 ratio. With a premium of

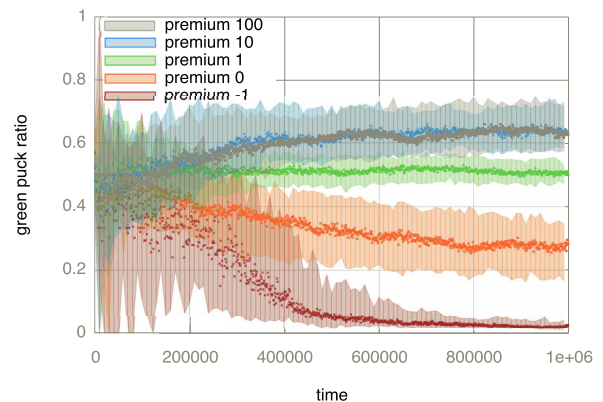


Figure 3: Development over time of the green puck ratio for a subset of the premium settings we considered. The points indicate the median green puck ratio for 1,000 time step intervals over 64 repeats, the shaded areas indicate lower and upper quartile.

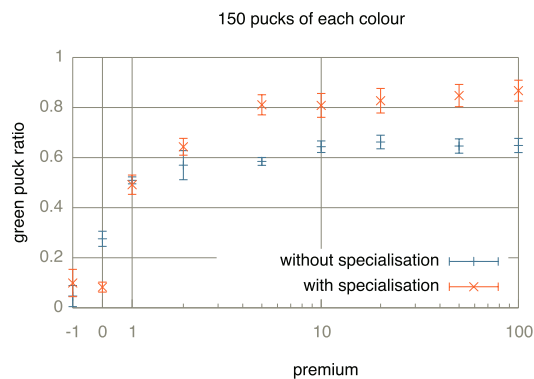


Figure 4: Mean green puck ratio in the final 1,000 time steps of runs with 150 pucks of each colour. The vertical bars indicate the 99% confidence interval over 64 repeats.

1 red and green pucks are equally valuable and the collective maintains this ratio. A premium of -1 has a profound impact: the robots learn to avoid green pucks and almost exclusively collect red pucks. This setting seems similar to the well-known poisonous food experiment, but the penalty for collecting ‘poisonous’ green pucks is effected during parent selection, not by the environment cutting short lifetime or reducing speed. A premium of 0 also leads to a substantial decrease in the green puck ratio: robots learn to focus on red pucks, but green pucks are not avoided and circa 30% of collected pucks is green. A premium of 10 increases the green puck ratio, which levels off around 0.65. The green puck ratio for premium 0 is in the same range as the red puck ratio for a premium of 10 (circa 0.3 and 0.35, respectively). This already indicates that larger premium values will do little to increase the green puck ratio (a premium of 0 for green pucks would have the same effect as a very high premium for red pucks).

This is borne out by the plot in Fig. 4, which shows the green puck ratio in the final 1,000 time steps of the simulation for varying premiums. We see that the green puck ratio among premium settings of 10 (or even 5) and higher barely changes. We also see that mono-skilled environments increase the impact of defining a premium: obviously, more robots will specialise in the higher rewarding task.

One reason for the lack of additional impact for higher premium values might lie in a saturation effect: if the robots simply cannot gather more green pucks than they do, the ratio can hardly improve. To test this hypothesis, we ran another set of experiments where there are 500 pucks of each colour. Figure 5 shows the results of those experiments. They show the same levelling off of premium impact, so it doesn’t seem to result from a saturation effect.

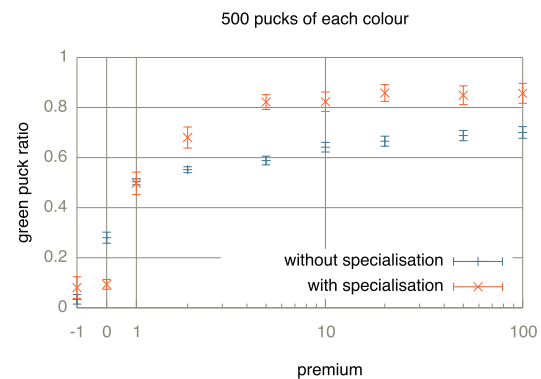


Figure 5: Mean green puck ratio in the final 1,000 time steps of runs with 500 pucks of each colour. The vertical bars indicate the 99% confidence interval over 64 repeats.

**Uneven Distribution of Pucks** We use a setting where there are more red than green pucks (150 vs. 50) as a proxy for having easy and hard tasks. In these experiments, the ‘natural’ green puck ratio is 0.25, which is what we see in Fig. 6 when the premium is set to 1 in a multi-skill environment. When the environment discourages generalists, the ratio is slightly lower because robots tend to specialise in the simpler task (earlier work showed that MONEE’s market mechanism plays a crucial role here (Haasdijk et al., 2013)). As was the case in the two scenarios where the puck distribution is balanced, increasing the premium past 5 or so has little further effect. The green puck ratio levels off between 0.3 and 0.4 for all premium values of 5 and greater.

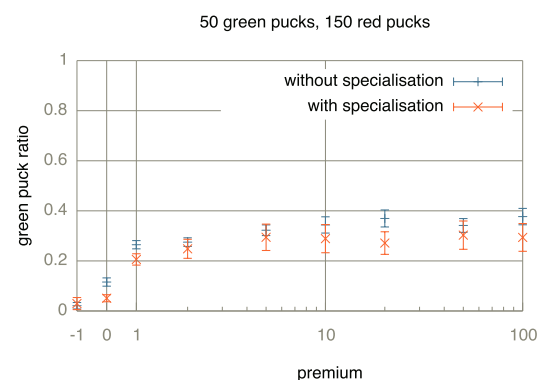


Figure 6: Mean green puck ratio in the final 1,000 time steps of runs with 50 green and 150 red pucks. The vertical bars indicate the 99% confidence interval over 64 repeats.

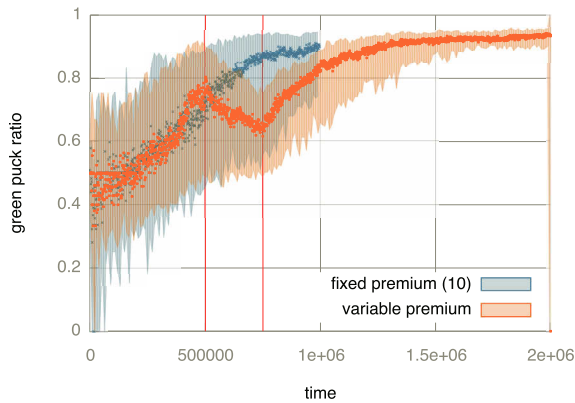


Figure 7: Development over time of the green puck ratio with changing premium with specialisation. The vertical red lines indicate when the premium is reset from 10 to 1 and back to 10. The points indicate the median green puck ratio for 1,000 time step intervals over 64 repeats, the shaded areas indicate lower and upper quartile. Green puck ratio for a constant premium of 10 shown for reference.

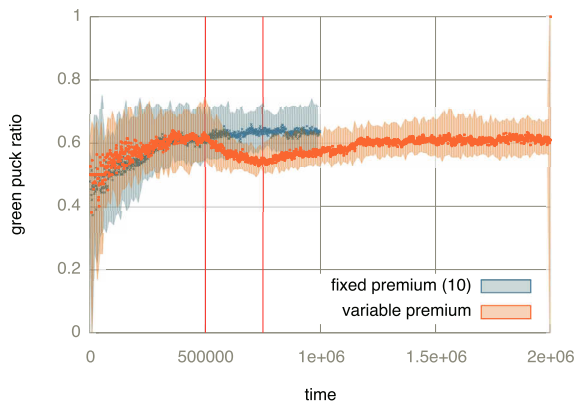


Figure 8: Development over time of the green puck ratio with changing premium without specialisation. The vertical red lines indicate when the premium is reset from 10 to 1 and back to 10. The points indicate the median green puck ratio for 1,000 time step intervals over 64 repeats, the shaded areas indicate lower and upper quartile. Green puck ratio for a constant premium of 10 shown for reference.

**Varying Premium Settings** Figures 7 and 8 show the puck ratio over time with changing premium with specialisation enforced and a multi-skilled setting, respectively. Note that we ran the experiment with changing premium for a further 1 million time steps to gauge long-term effects of changing premiums. With or without specialisation enforced, the green puck ratio initially develops unsurprisingly similar to

the control experiment (using a constant premium of 10). Then, as soon as the premium is set to 1 after 500,000 time steps, the green puck ratio drops to reflect the new prioritisation of tasks. When the initial premium is restored at 750,000 time steps, with or without enforced specialisation, the green puck ratio starts to rise again quickly and levels off where it was before the premium was changed. Hence, there is no memory effect when we change the premium in the course of a run, advocating for stable attractors that depend only from the premium value at hand. The change in puck ratio as a reaction to varying the premium is considerably more pronounced in the single-skill setting than when robots can collect both types of puck.

## Conclusions and Future Work

Experimental results on the effects of premium settings with the MONEE algorithm showed that setting premium values stand as an efficient mechanism to allow the user to control the prioritisation of tasks. On the one hand, setting negative premiums dramatically decreases the take-up of tasks. On the other hand, positive premiums enable to promote tasks, at least to some extent. Indeed, the relation between premiums and task distribution is not linear as the influence of increasing premium values is dampened after an environment-dependent threshold.

In the particular case of foraging with two kind of pucks, further experiments showed that controlling the evolution of a particular foraging behaviour is sensitive to the distribution of resources. Enforcing specialisation (i.e. penalising robots that forage both resources) can greatly increase controllability whenever both resources are available in equal amount, while dramatically decreasing controllability whenever an uneven distribution of resources is considered.

Lastly, controllability was also tested from the perspective of on-line tuning, i.e. changing premium values during the course of evolution to match user requests. Results revealed that premium values actually matched very stable attractors towards (expected) foraging behaviours.

Although the work presented here shows that collective foraging behaviour can be controlled to some extent through setting premium values, the non-linear (and thresholded) relation between premium values and task distribution remains to be further explored. We are indeed currently investigating the thresholding of the premium effect. Also, we are addressing the problem how to actually use premiums to automatically achieve a particular state of task distribution. To some extent, this is an inverse problem: while the desired task distribution may be known before hand, the method for tuning the premium values may well depend on the environment and the task at hand.

## Acknowledgements

The authors would like to thank Jean-Marc Montanier, Berend Weel and Nikita Noskov for many inspirational dis-

cussions on the topics presented here. We thank SARA Computing and Networking Services ([www.sara.nl](http://www.sara.nl)) for their support in using the Lisa Compute Cluster.

## References

- Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies – A comprehensive introduction. *Natural Computing*, 1:3–52.
- Bredeche, N., Montanier, J.-M., Liu, W., and Winfield, A. F. (2012). Environment-driven distributed evolutionary adaptation in a population of autonomous robotic agents. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):101–129.
- Bredeche, N., Montanier, J.-M., Weel, B., and Haasdijk, E. (2013). Roborobo! a fast robot simulator for swarm and collective robotics. *CoRR*, abs/1304.2888.
- Burtsev, M., Red'ko, V., and Gusarev, R. (2001). Model of evolutionary emergence of purposeful adaptive behavior: the role of motivation. In Kelemen, J. and Sosík, P., editors, *ECAL*, volume 2159 of *Lecture Notes in Computer Science*, pages 413–416. Springer.
- Haasdijk, E., Weel, B., and Eiben, A. (2013). Right on the money. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2013)*. To Appear.
- Jones, C. and Mataric, M. (2003). Adaptive division of labor in large-scale minimalist multi-robot systems. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1969 – 1974.
- Langton, C., editor (1995). *Artificial Life: an Overview*. MIT Press, Cambridge, MA.
- Langton, C. G. (1989). *Artificial Life: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Mascaro, S., Korb, K., and Nicholson, A. (2005). An alife investigation on the origins of dimorphic parental investments. In Abbass, H. A., Bossomaier, T., and Wiles, J., editors, *Advances in Natural Computation, Proceedings of the Australian Conference on Artificial Life (ACAL 2005)*, volume 3, pages 171–185.
- Menczer, F. and Belew, R. (1996). Latent energy environments. In *Santa Fe Institute Studies In The Sciences Of Complexity-Proceedings Volume-*, volume 26, pages 191–210.
- Menczer, F., Willuhn, W., and Belew, R. (1994). An endogenous fitness paradigm for adaptive information agents. In *CIKM Workshop on Intelligent Information Agents*. Citeseer.
- Nguyen, Q., Nguyen, X., O'Neill, M., and Agapitos, A. (2012). An investigation of fitness sharing with semantic and syntactic distance metrics. In Moraglio, A., Silva, S., Krawiec, K., Machado, P., and Cotta, C., editors, *Genetic Programming*, volume 7244 of *Lecture Notes in Computer Science*, pages 109–120. Springer Berlin Heidelberg.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA.
- Scheutz, M. and Schermerhorn, P. (2005). Predicting population dynamics and evolutionary trajectories based on performance evaluations in alife simulations. In Beyer, H.-G. and O'Reilly, U.-M., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2005)*, pages 35–42. ACM, ACM.
- Schwarzer, C., Höslér, C., and Michiels, N. (2010). Artificial sexuality and reproduction of robot organisms. In Levi, P. and Kernbach, S., editors, *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*, pages 384–403. Springer-Verlag, Berlin-Heidelberg-New York.
- Tang, F. and Parker, L. (2007). A complete methodology for generating multi-robot task solutions using asymptre-d and market-based task allocation. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3351 –3358.
- Ventrella, J. (2005). Genepool: Exploring the interaction between natural selection and sexual selection. *Artificial Life Models in Software*, pages 81–96.
- Walsh, W. and Wellman, M. (1998). A market protocol for decentralized task allocation. In *Multi Agent Systems, 1998. Proceedings. International Conference on*, pages 325 –332.
- Watson, R. A., Ficici, S. G., and Pollack, J. B. (2002). Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18.
- Wischmann, S., Stamm, K., and Wörgötter, F. (2007). Embodied evolution and learning: The neglected timing of maturation. In Almeida e Costa, F., editor, *Advances in Artificial Life: 9th European Conference on Artificial Life*, volume 4648 of *Lecture Notes in Artificial Intelligence*, pages 284–293. Springer-Verlag, Lisbon, Portugal.



# The relationship between Flocking Behaviour and the Emergence of Leadership

Francesco Pugliese<sup>1</sup> and Davide Marocco<sup>2</sup>

<sup>1</sup>Natural and Artificial Cognition Laboratory, University of Naples, Italy

<sup>2</sup>Centre for Robotics and Neural Systems, Plymouth University, United Kingdom

francesco.pugliese@unina.it, davide.marocco@plymouth.ac.uk

## Abstract

This paper examines the relationship between flocking behaviour and leadership. In order to achieve this aim, we simulate two co-evolving populations of robots: predators and prey. Behavioural and quantitative analysis indicate that a well-structured hierarchic leadership emerges in the population of predators after the evolution. The emergence of leadership relates to high levels of fitness, so leadership seems to be a winning strategy. We show that the leader role has been assumed by more explorative individuals. Moreover, exploratory behaviours mostly appear when there is a low following behaviour. Therefore, exploratory and following capabilities seem to be complementary both within every replication and within simulations with different perceptual conditions. On the other hand, leadership seems to be a strategy to enable followers to be more explorative.

Index Terms: Leadership, Evolutionary Robotics, Flocking

## Introduction

For the modern ethology and biology, groups of animals are autonomous units, enabling members to synchronize some activities, such as collective foraging and coordination in moving (Reebs, 2000). Specifically, the role of Leadership involves different degrees of conflicts. Across species, individuals are more likely to emerge as leaders if they have particular morphological, physiological, or behavioural traits increasing their propensity to act first in all the coordination problems. The consistent correlation between leadership and personality suggests the intriguing possibility that personality differences are maintained in populations, because they foster social coordination (King, et al., 2009). Many theoretical works have focused on how navigational information is exchanged between group members and how such information flow depends on the knowledge held by each member (Couzin, et al., 2005). In one study, the authors have examined the factors contributing to the formation of leadership/followership patterns in flocks of pigeons, focusing on the role of previous navigational experience (Flack, et al., 2012). The results prove that, in order to negotiate joint routes, pigeons make use of a complex decision-making system based on leadership mechanisms. Basically, less experienced birds are likely to follow more experienced conspecifics. All the pigeon groups exhibited a flocking behaviour. Flocking behaviour can be defined as the

capability of group's members to follow other individuals drawing those typical "lines", which are called "flocks". These behavioural patterns have been extensively identified by biologists and ethologists in the animal world: researchers tend to make distinctions between the "shoaling" behaviour of fish, the "swarming" behaviour of insects and the "herding" behaviour of land animals. Generally, flocking behaviour is used to identify groups of flying birds, the lines they trace are named "flocks" for this reason (Barnard, 1980). Recently, flocking has been simulated in many computer simulations with the aim of understanding the fundamental mechanisms (Kwasnicka, et al., 2007).

Researchers in robotics and agent-based modelling have usually focused on homogeneous groups. In one approach, they have evolved a team of four homogeneous robots for dynamically allocate roles through bodily and communicative interactions (Gigliotta, et al., 2009). In particular, evolved robots show a differentiation in both their communicative and non-communicative behaviours so that only one robot assumes the role of group leader. In another experiment, a group of agents were simulated for the task of reaching a target in a two dimensional environment (Gigliotta, and Miglino, 2007). Lastly, some researchers have evolved a robot colony to study the possibility for the evolution of leadership patterns (Lee, et al., 2011). In this work each robot has a prearranged social position, such as, leader, follower, and stranger.

In the present paper we discuss an experiment focused on spontaneous leadership emergence mechanisms (namely without any prearrangement of the social roles). The Robots were evolved by the use of Evolutionary Robotics techniques. This experiment has a two-pronged value, one for robotics, one for social science. In robotics: the genetic differentiation of robots' control systems could contribute to build a new generation of autonomous robots with a leadership/followership hierarchic structure needed for navigational tasks in an undiscovered environment. For social sciences and artificial life, it may be possible to answer some interesting questions related to leadership, such as: Is leadership unavoidable for a social decision-making problem? What are the characteristics and skills of a leader? How environmental and individual characteristics affect the emergence of leadership? What is the ratio between the leaders' portion and followers' portion in a group? The final two questions would be: What is the relationship between flocking behaviour and leadership patterns emergence? Does

leadership arise in any flocking groups or flocks without leaders could exist?

## Experimental Setup

### The Task

A group of 40 simulated robots live in an environment consisting of a 550cm x 550cm squared arena surrounded by walls. Each robot is inspired by the Khepera Robots bodies, which have circular chassis with a diameter of 5.5cm. The robots' bodies are equipped with visual sensors and two wheels by which the robots move in the environment (see Figure 1). Figure 2 depicts a schematisation of the experimental setup. The environment contains 20 predator robots and 20 prey robots. The only physical difference between the predators and the prey is the colour: blue for predators and green for prey. Both predators and prey are evolved using Evolutionary Robotics methodology (Nolfi, and Floreano, 2000). When a predator robot bumps against walls or against another predator robot, it bounces back in the neighborhood of the contact point facing a new (i.e. randomly chosen) direction.

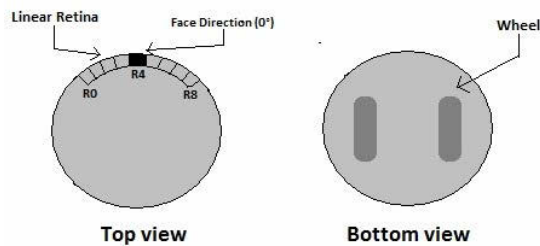


Figure 1: Schematisation of top and bottom view of the robot chassis.

These bumping rules are followed by the prey robots too, with the exception of bumping into the predators. In fact, there are further behavioural difference between predators and prey: whenever a predator's body approaches and touches a prey's body, the prey disappears, meaning that the predator eats it and the prey consequently dies. On the other side, predators cannot die, in this model. Another substantial difference between prey and predators consists of the different fitness function (which will be illustrated in the next paragraph). The vision system of both prey and predators is based on a linear retina made of 9 photoreceptors (R0-R8) that perceive gray scaled colours. The field of view (FOV) of each robot is 90 degrees wide and represents the extent of the observable world that the robot is able to see at any moment. The FOV ranges from -45 degrees to +45 degrees with respect to the face direction (0°), which is the robot's moving direction. In this way, each photoreceptor manages a 10° wide portion of the FOV: the first photoreceptor is associated to a range of [-

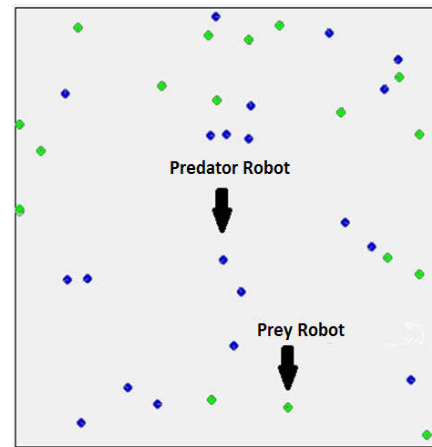


Figure 2: The environment and the robots.

45°,-35°] with respect to the direction faced, the second one to [-35°,-25°], and so on.

When an object (for instance another robot) is located in front of a photoreceptor (within its own vision angle), it is activated to a value encoding the colour of the object. Perceived colour values are grey-scaled by the retina system and normalised in the range [0,1]. Therefore, the prey's green colour activates the photoreceptors at 0.26, which is the normalised value relating to the gray scaled green. The predators' blue colour enables photoreceptors at 0.97. The maximum vision distance for each retina sensor is 55cm. So, if an object is further from a photoreceptor than 55cm, it cannot be detected.

### Neural Controller

An Artificial Neural Network (ANN) controls the behaviour of each robot. The neural network consists of 3 layers with 13 neurons in total: each neuron is connected to the other layers with no recurrent connections. This feed forward topology is schematised in Figure 3. The input layer contains 9 neurons which encode the output from the 9 retina's photoreceptors. In other words, input units receive values (normalised in a range between 0 and 1) from the retina's sensors depending on the gray level of the perceived image. The hidden layer consists of 2 units, and the output layers are the controllers for the motor units: output neurons encode the speed of two wheels which enable the robot to move within the environment. The activation of all the network's units are in the range [0,1]. Internal and output neurons are characterised by a sigmoid activation function (logistics).

### Artificial evolution

The evolutionary process for the robots is based on a ranking type genetic algorithm. Each individual is identified by a genotype that encodes the neural network's parameters. These encoded parameters represent the synaptic connection weights and biases.

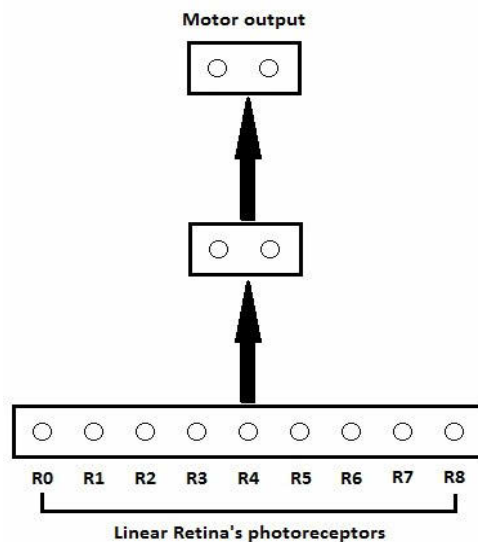


Figure 3: The control system of predator and prey robots.

Furthermore, initial parameters are randomly generated in the range  $[-5, +5]$ . Lastly, each parameter is encoded as a sequence of 8 bits. Thus, initially, the environment is populated by a generation of completely “naive” robots (namely, with a randomly generated genome) with no skills on how to move and detect the food sources. In each generation, 40 individuals are inserted into the environment and left to act randomly there. The originality of the algorithm is that individuals are evaluated all together in the environment. In this way, all the robots present in the environment (both prey and predators) in a given moment are characterised by a different genotype that makes them unique in the population (genetic heterogeneity). At the end of each generation, a different ranking and mutation process is applied to both prey and predators, in order to simulate two different species. Each generation is made of 20 epochs. At the beginning of each epoch, every robot starts from random positions. The life time consists of 3,000 time steps. At the end of their life, all the 20 predators are ranked according to the average number of prey eaten in all the epochs. Each of the 4 higher-ranked predators generates 5 offspring which inherit the genotype of their father. The first preserves its father’s genotype entirely (elitism) whereas the rest of the offspring’s genomes receive a random mutation with a rate of 2%. The total number of new predators ( $4 \times 5 = 20$ ) populates the next generation. Similarly, the 20 prey robots are ranked separately. All the evolutionary process carries on for 300 generations. All the simulation is repeated for 10 replications (or seeds).

The Fitness function is computed differently for predators and prey: when a predator bumps against a prey robot, the prey disappears from the environment (i.e. it is dead) and the predator’s fitness score is increased by a value of +1.0.

Each predator robot always lives 3,000 time steps. Whereas each prey robot can die at any time, so prey can have a shorter life span than the predators. A prey’s fitness is calculated by the number of time steps in which it can survive.

## Results

After the evolution, we observed that the predators evolve the ability to run after the prey, and preys evolve the skill of escaping from predators. Moreover we have noticed the emergence of a flocking behaviour between the predators. On the other hand, the prey do not display any specific grouping behaviour, they just tend to explore the environment. The prey simply avoid the predators when they are in the neighbourhood.

Average predators’ fitness curves reveal a constant trend for the best and average populations, as showed in Figure 4.

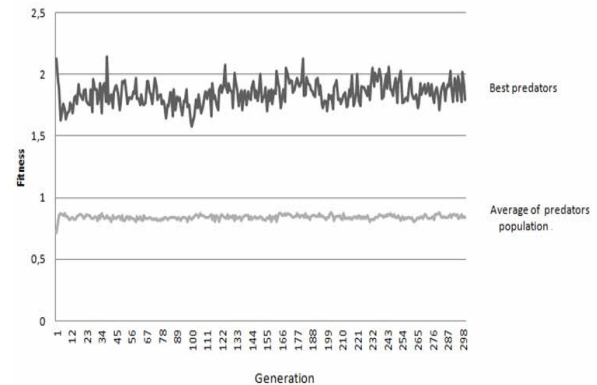


Figure 4: Visualisation of the average of all 10 predators’ fitness curves, bests (black) and averages (light grey).

The steadiness of fitness curves is present for both predators and prey. In spite of fitness constancy, predators and prey improve their skills and performances throughout the generations. This effect has been explained in other related works by the “arms-races” effect (Nolfi, and Floreano, 2006).

In practice, arms-races may emerge in every situation where a co-evolution of two species is present. That is why, in our simulation, fitness curves appear stable. Nevertheless, robots’ strategies and skills improve and become more efficient during the evolution: predators become faster to hunt prey and prey become smarter to avoid predators. Another factor that makes predators’ fitness curves constant in time, is the fact that, in each generation, only 20 prey can be eaten in total, because the prey will not be replaced in the environment after they die.

To find a single indicator on the fitness reached by robots in each replication, we have calculated the average of average fitness over the last 20 generations (Fitness Indicator).

Apparently there is an unexpected inter-replication variation of fitness. So the first question we have tried to answer is: What is the phenomenon behind the substantial inter-seed variation of average fitness? In order to understand the reason of this variation, we have tried to calculate a static aggregation measure of the predators’ populations in the ecological environment. From this point we only consider the predators’

population for further analysis, as there are no interesting emerging social behaviours in the prey population for our aim. The “Aggregation Measure” has been calculated by measuring the distance between each robot and the nearest robot, in each time step, for the last generation’s populations. All the time steps and epochs measures have been averaged. The lower values correspond to more aggregation and, vice versa, higher values correspond to less aggregation.

By running a correlation between the Fitness Indicator and the Aggregation Measure, the Pearson’s Correlation Coefficient between those two series of data is  $\rho = -0.7$ , which indicates high reverse-correlation between Fitness and Aggregation.

This means that the higher the aggregation, the higher the fitness. In Figure 5 all the series are reported on a plot with the correlation coefficient.

To better understand the mechanisms underpinning the flocking, we have tested each single predator in a reduced environment called “Laboratory”. Laboratory is a square with a size of 150cm x 150cm. Firstly, we have inserted a single predator into the environment, and we have plotted the individual trajectory, as illustrated in Figure 6a.

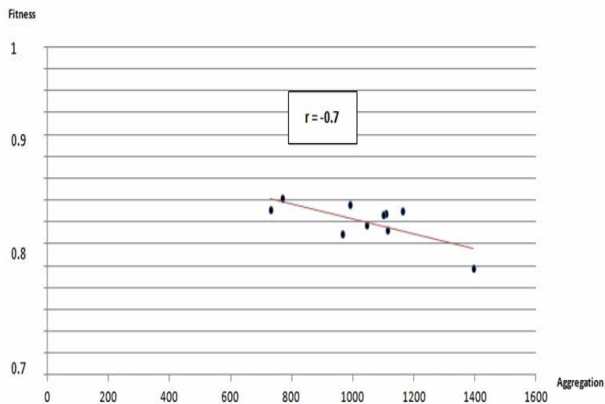


Figure 5: Correlation between Aggregation Measure and Fitness Indicator.

Some robots display a small exploratory ability, others a medium exploratory ability, and other robots have a large exploratory ability. By placing the robots side by side in the same environment, we observed the behaviour illustrated in Figure 6b. We have noticed that, almost in every couple, one robot always leads and one robot follows. From these observations it appears that the flocking seems to be regulated by a hierarchy among the predators and this is predetermined in advance by the evolution, this hierarchy is numerically proved by exploration measures illustrated later on. We have supposed the hierarchy is guaranteed by every single trajectory (i.e. by every single exploratory ability). On the other hand, the hierarchy cannot be regulated by the colour, because all the 3 robots have the same colour. Exploiting the information from retina photoreceptors, each robot is able to discern the angle of another robot’s movements. In other words, each robot is able to discriminate the arching amplitude of the curvilinear trajectories that another robot is

able to draw. By this amplitude information, robots can recognize the hierarchic degree of their partners.

Thus, every robot is capable to decide whether to follow or to lead, by using this strategy. The leadership appears to be “relative”, namely each leader is not an absolute leader but may be a follower of another robot.

To clarify and support these hypothesis returned by the behavioural analysis, we have developed a series of analytical measures. The first Measure, that we have conceived, is called “Leadership Measure by Vision”, which measures each predator’s leadership hierarchical rank by exploiting the vision system. We have inserted all the possible predator couples into the Laboratory Environment. Each of the 20 predators have been paired with each of the 19 others. For each sub-test, only 2 robots are present in the environment, in the same time. Then, we have counted in how many time steps each predator sees something, namely how many time steps at least one retina photoreceptor is activated. The hypothesis is that if there are only two individuals in the same environment, the leader will see less than the follower, if there are no other objects.

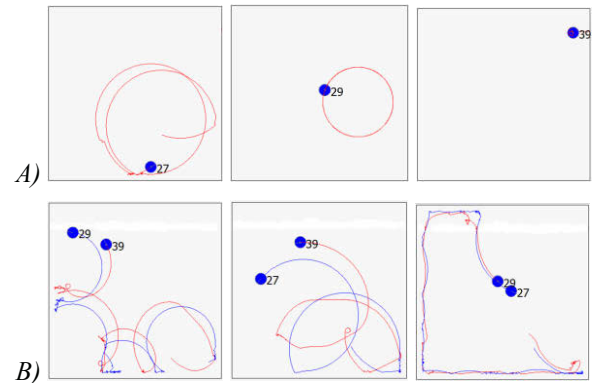


Figure 6: (A) Trajectories of some predator exemplars. They are predator number 27, 29 and 39. Trajectories of predators couples.

In fact, the leader should more likely be at the head of the line whereas the follower should be on the tail.

Correlation Coefficient between “Aggregation” and “Leadership Measure by Vision over replications” has returned a  $\rho = -0.8$  that proves a strong reverse-correlation between leadership and aggregation. This correlation is shown in Figure 7. This means that the higher the leadership, the stronger the aggregation in the group. The reverse-correlation appears because of the design of the leadership measure: the higher the leadership measure, the lower the vision value.

We can argue that, if the aggregation correlates with the fitness, and fitness correlates with leadership, then the leadership correlates with the fitness. That is, high levels of fitness correlate with high levels of leadership

Another interesting issue is in which way is the leadership role connects with the exploratory ability. Predator robots seem to display different exploratory skills. Hence, we have calculated the exploratory ability for each single robot.



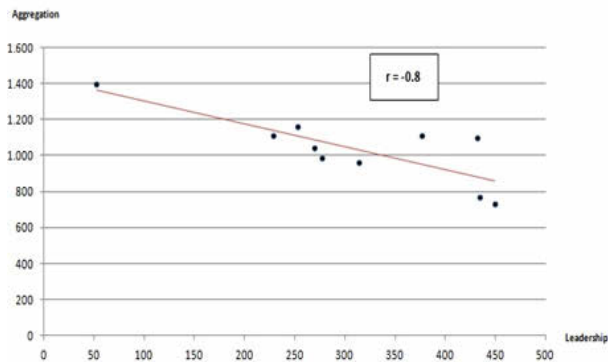


Figure 7: Correlation between Aggregation Measure and Leadership Measure.

Each test has been performed on the last generation's population of predators for 20 trials lasting 3,000 time steps. Each value has been averaged over all the trials and reported on the bar-plot. The ability to explore has been then related with the ability of the predators to follow other robots in the environment.

To this end we have identified a "Exploratory Ability Measure" which is depicted in Figure 8, dark grey bars and a "Followership Measure". We have measured the exploratory ability of each robot in ecology by counting how many 5.5cm x 5.5cm sized cells each robot visits only once. (we call ecology the evolutionary environment to distinguish it from the smallest laboratory environment) in two different conditions: *vision* condition, in which the robot can see any other robots in the environment and *no-vision* condition, in which a robot can only see the prey and it is blind to the other presence of the other predators in the environment.

This test has been executed, both conditions, on the last generation's predators for 20 trials.

We noticed that, the *vision* condition produces an increase of exploratory abilities, especially in those cases where robots were not good at exploration in the *no-vision* condition. The increase of exploratory ability has been schematised in Figure 9. As we can observe in the *vision* condition the less explorative robots became more exploratory. It may indicate that the ability to follow other predators in the group seems to be a mechanism to make the entire group able to become more exploratory with respect to the situation in which they are alone. This could arguably be an effect of the social relationship in the group. We can then suppose that, in those situations where there are many individuals who are not genetically predisposed to assume exploratory behaviours, leadership may facilitates the group cohesion and performance by increasing the exploration.

Another interesting insight derives from measuring the exploratory gap between the *no-vision* condition and the *vision* condition, replication by replication. Essentially, we have averaged all the values of the *no-vision* exploratory ability over replications.

A gap appears between replications (light grey bars): the gap can be regarded as the average ability "to follow" of the robots in one replication.

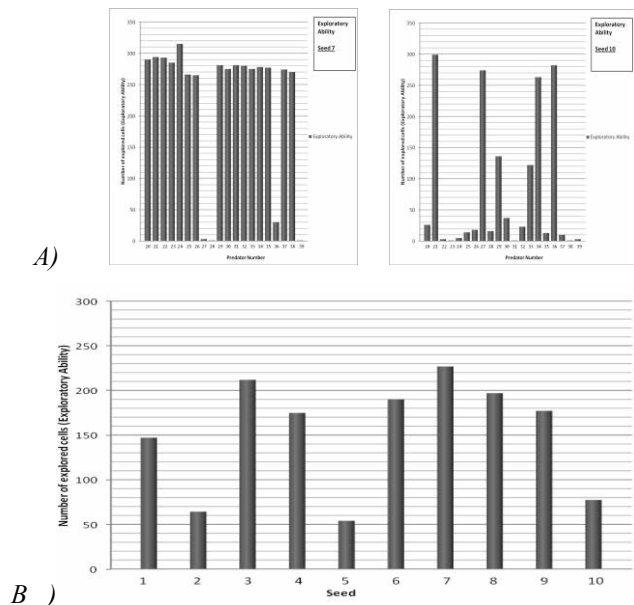


Figure 8: (A) Exploratory Ability Measure. In the picture there are the values of Replication no.7 and Replication no.10. (B) Exploratory Ability Measure over replications (average).

Indeed, we can consider that, from the *no-vision* to *vision* condition, each robot gains an increase of their exploratory abilities, which is directly proportional to the propensity of the robot to follow someone else. If we "isolate" this gap, we acquire a measure of robots' following abilities over replications: the "Followership Measure".

By calculating the Pearson's correlation coefficient between Leadership Measure and Followership Measure the value is  $\rho = -0.79$  confirming a strong correlation.

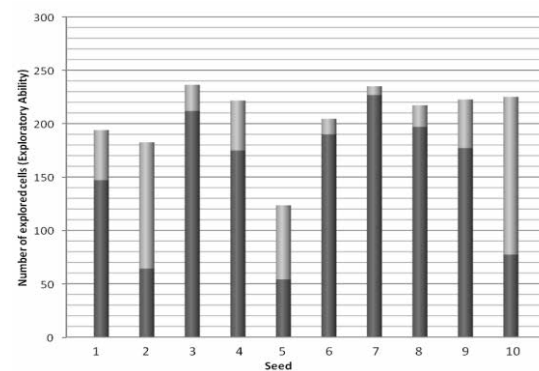


Figure 9: Exploratory Ability Measure in *no-vision* (dark grey) and *vision* condition (dark + light grey). The light grey values indicate the Followership Measure.

In Figure 10 this correlation is graphically visualised. This indicates that the stronger is the leadership in a replication, the stronger is the followership in the same replication, as indication of the fact that leadership only emerges where the social group is based on a clear leader-follower organization and someone leads and other follows.

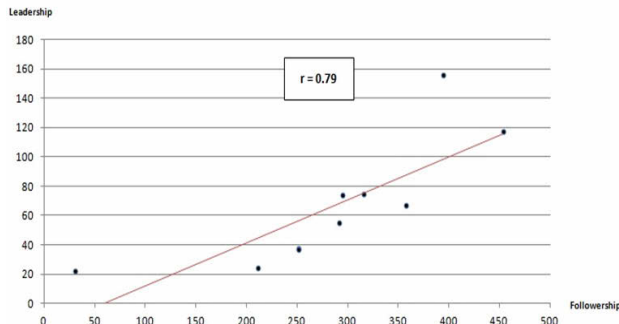


Figure 10: Correlation between Followership Measure and Leadership Measure.

The motivation for the flocking emergence is determined by the fact that every predator robot is characterised by a maximum limit of vision distance (55cm), when predators cannot see any prey, they tend to follow another predator rather than doing nothing. This fact is proved in Figure 11 where a chart shows that increasing the vision distance, the following behaviour decreases and vice-versa. The specificity of flocking behaviour is also suggested by the fact that following another predator is notably different than hunting a prey. In fact, when predators move after another con-specific, they do not tend to bump against it, but they just limit themselves to follow keeping a safe distance. Instead, hunting consists of following the prey until the predator reaches it and bumps against it, in order to eat the prey. A careful analysis of the exploratory and following abilities, by means of previous charts, shows another interesting piece of information: the exploratory ability and following ability are reciprocally complementary. This means one ability excludes the other one. For example, in the seed 7, all the predators appear to be explorers rather than followers, whereas in the seed 10 they display an inclination for following rather than exploring. For this reason, we have implemented an analysis of exploration and following abilities depending on different perceptual conditions. In substance, we have re-evolved the robots with different vision conditions, namely by varying the vision distance limit: 13.75cm, 27.5cm, 41.25cm, 55cm, 82.5cm, 165cm and 220cm. We have not been able to sample many more vision conditions because of the elevated computational and time costs of each single evolution. Anyway, the number of completed samples has seemed to be sufficient for the present. The limit of 55cm has been used as the default condition, because it was adopted in the initial evolution. Therefore, we have considered the condition “55cm” as baseline for the comparisons. Again, every simulation has

been evolved for 300 generations and for 10 replications in every vision condition. When all the evolutions have been accomplished, we have calculated the average of the “Exploratory Ability Measure over replication” and of the “Followership Measure”, for each condition.

The result for the “Exploratory Ability” and “Following Ability” through different vision conditions is depicted in Figures 11. We have graphically interpolated all the points in order to highlight the data trend.

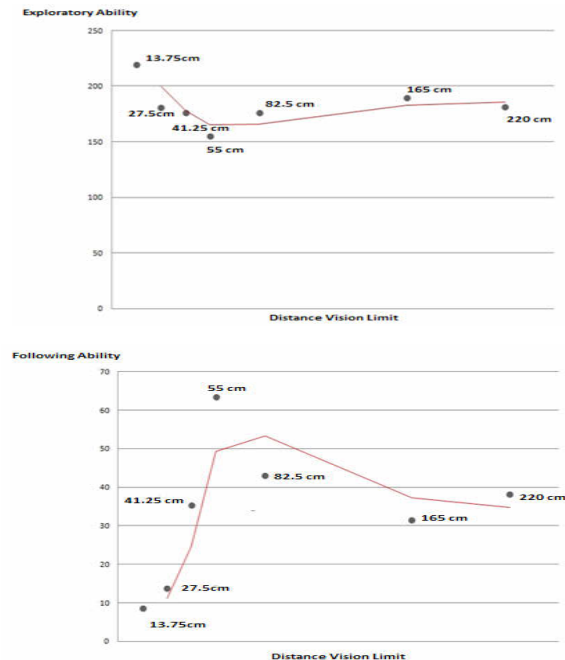


Figure 11: (A) Exploratory Ability through different Distance Vision Limits. (B) Followership Ability through different Distance Vision Limits.

## Conclusions

In conclusion, the experiment reported here indicates that in a population of two co-evolving species of robots, with a genetically variable distribution of skills, flocking and leadership are often observed. Although the fitness keeps constant over the generations for the “arms-races” effect, each species’ skills appear enhanced at the end of the evolution: predators are better at hunting prey and prey are better at escaping. An inter-replication variation is present in the indicators of predators’ fitness and aggregation, which underline a different “social” organisation among the predators that we interpret as leadership, which is mainly due to different initial genetic traits (which are randomly selected). In replications where there is a strong component of leadership, there seems to be a stronger aggregation. Furthermore, a strongly structured hierarchy appears in the predator’s population: the rank of each robot is regulated by the explorative attitude of each robot (namely, the amplitude

of movements). Fitness, leadership and followership measures are in strong correlation.

In this way, we guess it is possible to conclude that a group of artificial agents exploit leadership/followership patterns to solve the task of exploring the environment and moving collectively toward the position of the prey. These leadership patterns correlate with higher fitness, which suggests leadership is a winning strategy. In other words, a “peer-to-peer” flocking behaviour is not enough to guarantee a smart movement of the group, but the emergence of leadership is needed for achieving better performances.

Another interesting result is that all the exploratory individuals do not tend to be good followers and vice versa. This indicates that there is a specialisation of skills in populations, according to different simulated conditions. This suggests a theoretical limit: exploration and following are two complementary skills.

Other interesting future directions could be investigating if the bigger the group size, the smaller the leaders portion. Furthermore, some improvements might be achieved in this simulation by examining, in depth, some of the unclear aspects such as the correlation between leadership emergence and fitness and the relationship between genetic variability and leadership emergence.

## References

- [1] Reeb, S.G. (2000), “Can a minority of informed leaders determine the foraging movements of a fish shoal?” *Anim. Behav.* 59, 403–409.
- [2] King, A. J., Johnson D.P. D., Dominic D.P. Johnson and Van Vugt, M., “The Origins an Evolution of Leadership. *Current Biology*”, 2009.
- [3] Couzin ID, Krause J, Franks NR, Levin SA., “Effective leadership and decision-making in animal groups on the move”. *Nature*. 2005 Feb 3;433(7025):513-6.,2005.
- [4] Flack, A., Pettit, B., Freeman, R., Guilford, T. and Biro, D., (2012), “What are leaders made of ? The role of individual experience in determining leader-follower relations in homing pigeons”, *Animal Behaviour*.
- [5] Barnard, C.J., (1980), “Flock feeding and time budgets in the house sparrow (*passer domesticus* l.)”, *Animal Behaviour*, 28, 295-309.
- [6] Kwasnicka, H., Markowska-Kaczmar, U., Mikosik, M., (2007), “Open-ended Evolution in Flocking Behaviour Simulation”, *Proceedings of the International Multiconference on Computer Science and Information Technology* pp. 103–120.
- [7] Gigliotta, O., Miralli, M., and Nolfi, S., (2009), “Who Is the Leader? Dynamic Role Allocation Through Communication in a Population of Homogeneous Robots”, In Serra R., Villani M., Poli I. (Eds.): *Artificial Life and Evolutionary Computation. Proceedings of Wivace 2008*. Singapore, World Scientific: 167-177,2009
- [8] Gigliotta, O., and Miglino, O., (2007), “Groups of Agents with a Leader”, *Journal of Artificial Societies and Social Simulation* vol. 10, no. 41
- [9] Lee, Seung-Hyun, Si-Hyuk Yi, and Sung-Bae Cho, (2011) "Emergence of Leadership in Evolving Robot Colony." *Neural Information Processing*. Springer Berlin/Heidelberg.
- [10] Van Vugt, M., (2008), “Follow me”, *New Sci*, 198, 42–45.
- [11] Nolfi, S. and Floreano, D. (2000), “Evolutionary Robotics: The Biology, Intelligence and Technology of Self-Organizing Machines”, MIT Press, Cambridge.
- [12] Nolfi, S., Floreano, D., (2006), “Coevolving Predator and Prey Robots: Do “Arms Races” Arise in Artificial Evolution?”, *Mit press Journals*.

# Learning Schooling Behavior from Observation

Brian Hrolenok and Tucker Balch

Georgia Institute of Technology, Atlanta, GA 30332  
 {bhroleno, balch}@cc.gatech.edu

## Abstract

Agent-based simulation is a valuable tool for biologists studying animal behavior, however constructing models for simulation is often a time-consuming manual task, and validation of these models requires a principled approach. We present a framework for using machine learning techniques to automatically construct behaviors from tracking data of live animals from video that can be run in a simulated environment. Using this framework, we provide results for automatically learning the schooling behavior of *Notemigonus crysoleucas*.

## Introduction

The motivation for this work has been to enable the work of biologists that study collective behavior through agent-based models. Agent-based models have been successful in analyzing the behavior of social insects such as ants and bees (Pratt et al., 2005; List et al., 2009), although currently such models are constructed after manual processing of video of the collective behavior of the animal. An automated method for constructing these models would enable more rapid iterative refinement of biological theories by allowing researchers to test hypotheses *in silico* with parameters that would be difficult to manage in real animals, as well as provide a tool for performing principled validation as outlined by Yang et al. (2012).

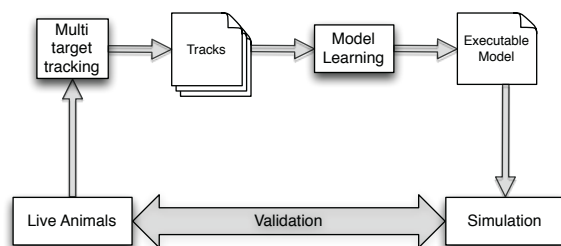


Figure 1: Workflow for automatically constructing executable behaviors from observation

The manual process of model creation usually consists of frame-by-frame annotation of video of the animals in question and statistical analysis of the resulting data, and the automation of this process can be decomposed into two corresponding subproblems: multi-target tracking of animals in video, and learning an executable model from those tracks. This workflow is outlined in Figure 1. The computer vision community has developed a number of algorithms for solving the multi-target tracking problem in specific domains, including tracking biological agents such as humans and ants (Feldman et al., 2012). Given a tracking algorithm that can produce tracks of individual agents with reasonable accuracy, the task is then to construct an executable agent-based model of behavior from the given data.

## Learning fish schooling behaviors

The schooling of *Notemigonus crysoleucas* is an interesting collective behavior, one example of many types of “flocking” behavior found in nature. While the motion of the group as a whole is generally very complex, Reynolds (1987) has shown that individuals following fairly simple local rules can result in global flocking behavior. If we can then correctly learn a model of how the fish react to the features of their local surroundings, we should be able to reproduce the global schooling behavior by simulating fish in a similar environment that react according to the learned model. This means we need to identify which features of the environment the fish are reacting to, compute those features for each track in the tracking data, learn a mapping from features to reactions, and compute the identified features as part of the simulation.

## Fish sensor features

There are several important features of the environment that effect how individual fish act as part of a school, and how the schooling phenomenon arises in groups of fish. We took inspiration from both classic flocking literature (Reynolds, 1987) and more recent work by Katz et al. (2011) in determining which features to include. From the collected tracking data we compute 13 features: 8 proximity sensors, the



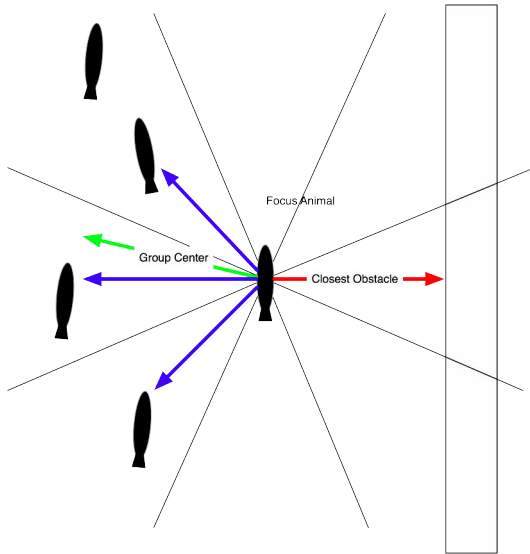


Figure 2: Sensor model for *Notemigonus crysoleucas*

$x$  and  $y$  components of the normalized vector to the school center, the  $x$  and  $y$  components to the nearest obstacle, and a binary (near-far) distance to the school centroid that is one if the fish is within 3 body lengths of the school centroid and zero otherwise. The proximity sensors are thresholded at 4 body lengths, and the obstacle vector and school centroid calculation are both limited to objects within 1m.

These specific features can be thought of roughly corresponding with three of the classic components proposed by Reynolds: the 8 proximity sensors are useful for determining *separation*, the obstacle vector provides a mechanism for avoiding environmental obstacles, and the group center vector influences *cohesion*. Notice that we do not include any alignment term, and as Katz et al. suggest, the apparent group alignment is an emergent phenomenon, not a determining feature that the individual fish react to.

### Fish actuators

In order to learn how the fish should react to a given feature vector, we must also quantify how the observed fish actually moved in response to the computed features. The tracking data includes the position  $(x, y)$  and orientation  $(\theta)$  of each fish at each time step (see Figure 3). From consecutive time steps, we can calculate the change in position and orientation as a rough estimate of the velocity of the fish in reaction to its local environment, as long as the time interval is relatively short (the tracking data we use is computed frame-to-frame from video running at 30Hz).

### Learning

Using the paired feature vector and velocity estimate as training data, we can construct a  $k$ -NN which maps any new feature vector to the  $k$  most similar training instances and

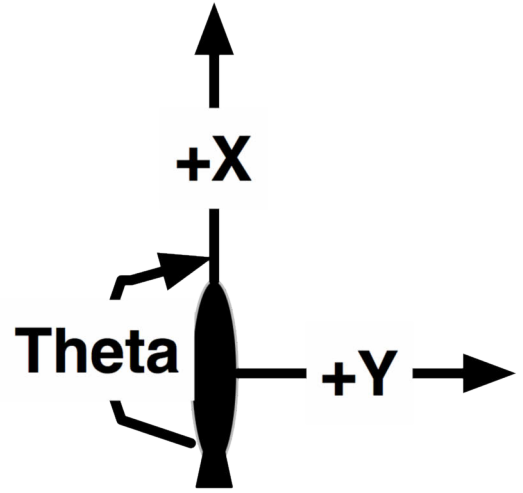


Figure 3: Actuator model for *Notemigonus crysoleucas*

the associated observed motions. One interesting difference from the standard  $k$ -NN in this instance is that the output associated with each feature vector is a continuous set of values describing how the fish moved, rather than a discrete class. In the standard  $k$ -NN with discrete output, each of the  $k$  nearest neighbors to a given query  $q$  votes for one of the possible discrete outputs and the output with the highest number of votes is returned as the class for the query. We can generalize this by returning the output of an arbitrary function  $g$  of the  $k$  nearest neighbors for a given  $q$ :

$$f(q) = g(\{q_i \mid d(q_i, q) \leq d(q_j, q), \forall i < j, i = 1 \dots k\})$$

where  $d(q_i, q)$  is the distance between  $q$  and  $q_i$ . In the standard  $k$ -NN, the function  $g$  just returns the class with the maximum number of votes, or the mode of the classes of the  $k$  neighbors. Other choices for  $g$  include the mean, or median. Empirically, we've found that sampling randomly from the  $k$  neighbors works better than taking the mean. This might be due to the fact that the animals do not behave in a completely deterministic manner: In the case where the fish is approaching a wall head-on it may turn left or right to avoid it, but the average of both cases would be to head straight forward, leading to a collision with the wall. On the other hand, sampling randomly from the neighbors would produce both left and right turns, and in the proportion that they are represented in the data.

### Simulations of learned behavior

For our training set, we used tracking data collected from a 54 minute video of 30 *Notemigonus crysoleucas* schooling in a shallow tank 2.1 meters long by 1.2 meters wide<sup>1</sup>. From

<sup>1</sup>This data was one replicant from the experiments performed in Katz et al. (2011).

the tracks we computed the 13 features and 3 velocities described previously. The collected data amounted to roughly 2.6 million input/output pairs, which we used as training data for a  $k$ -NN. We constructed a simulation with 30 fish in a similar environment using BioSim, a freely available simulation toolkit<sup>2</sup>. At each time step, each fish computes the 13 features described earlier and sets its velocity by selecting from the  $k$  nearest neighbors. Figure 4 shows screenshots from the simulation and the resulting schooling behavior.

The fish are initially placed in the environment at random locations moderately spread out, but they quickly form into a single dense school. The school tends to stay close to the boundaries, tightly clustered. This is very similar to the behavior of the real fish in the training data, as shown in Figure 5.

As discussed in our motivation, one reason such agent-based simulations are useful is the ability to run experiments in simulation that would be difficult or time consuming to perform using live test animals. To illustrate this capability, we ran a simulation of 300 fish in a larger (3m by 5m) tank.

Figure 6 shows a screenshot of the 300 fish simulation. Notice how the fish have separated into several distinct schools.

## Conclusion and Future work

This work has illustrated how the process outlined initially can be applied to learn the schooling behavior of fish from video: by applying a standard multi-target tracking algorithm to video to produce tracks of position and orientation, then computing a set of input/output (features/motions) pairs from the tracking data, then using those pairs as training data for a learning algorithm ( $k$ -NN) to construct a mapping between observed features and agent output, and finally using that mapping as the basis for a simulation. Our experimental results show that the collective behavior of agents following the learned behavior is qualitatively similar to the schooling behavior which generated the training data.

It's important to note that the choice of algorithm for both the tracking and learning components are crucial. The noise inherent in the tracks produced by the tracking algorithm must be relatively small, otherwise the training data used by the learning algorithm may be so noisy as to not permit an accurate mapping. The tracking algorithm must also be able to account for all the variables of interest, such as orientation. The choice of learning algorithm also has a profound effect. In the case of schooling fish, it is apparent from flocking models that the collective schooling behavior can arise from purely local and *reactive* rules. In other words, the mapping we've discussed so far is *stateless* in that the output is dependent only on the observed features, and not any internal memory or state. However there are many interesting types of behavior that are not stateless in this sense,

such as foraging in ants (Yang et al., 2012) or the honey bee "waggle dance" (Oh et al., 2005). Learning these types of behaviors requires an algorithm that can handle state such as presented by Balch et al. (2006), and such algorithms are a focus of our current and future work.

## Acknowledgements

We would like to thank Iain Couzin and the Collective Animal Behavior lab at Princeton University for the use of their tracking data.

## References

- Balch, T., Dellaert, F., Feldman, A., Guillory, A., Isbell, C. L., Khan, Z., Pratt, S. C., Stein, A. N., and Wilde, H. (2006). How multirobot systems research will accelerate our understanding of social animal behavior. *Proceedings of the IEEE*, 94(7):1445–1463.
- Feldman, A., Hybinette, M., and Balch, T. (2012). The multi-iterative closest point tracker: An online algorithm for tracking multiple interacting targets. *Journal of Field Robotics*, 29(2):258–276.
- Katz, Y., Tunström, K., Ioannou, C. C., Huepe, C., and Couzin, I. D. (2011). Inferring the structure and dynamics of interactions in schooling fish. *Proceedings of the National Academy of Sciences*, 108(46):18720–18725.
- List, C., Elsholtz, C., and Seeley, T. (2009). Independence and interdependence in collective decision making: an agent-based model of nest-site choice by honeybee swarms. *Philosophical transactions of The Royal Society B: biological sciences*, 364(1518):755–762.
- Oh, S. M., Rehg, J. M., Balch, T., and Dellaert, F. (2005). Data-driven mcmc for learning and inference in switching linear dynamic systems. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 2*, pages 944–949. AAAI Press.
- Pratt, S., Sumpter, D., Mallon, E., and Franks, N. (2005). An agent-based model of collective nest choice by the ant *Temnothorax albipennis*. *Animal Behaviour*, 70(5):1023–1036.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics*, volume 21, pages 25–34. ACM.
- Yang, Y., Quitmeyer, A., Hrotenok, B., Shang, H., Nguyen, D., Balch, T., Medina, T., Sherer, C., and Hybinette, M. (2012). Ant hunt: Towards a validated model of live ant hunting behavior. In *Twenty-Fifth International FLAIRS Conference*.

<sup>2</sup><https://github.com/biotracking/biosim2>



Figure 4: Simulated fish at consecutive intervals. The fish have a strong tendency to stay with the school, and congregate near the walls of the tank much like the real fish.



Figure 5: Replayed tracking data of real fish.





Figure 6: Simulation of 300 fish in a large tank using the same 30 fish training data

# Evolution of Mutual Trust Protocol in Human-based Multi-Agent Simulation

Hirotaka Osawa<sup>1</sup> and Michita Imai<sup>2</sup>

<sup>1</sup>University of Tsukuba

<sup>2</sup>Keio University

osawa@iit.tsukuba.ac.jp

## Abstract

Acquisition of the opponent's model and achieving mutual trust with others are notable traits of humankind's intelligence. Achieving mutual trust is a big challenge for artificial intelligences, and it is a key factor in trading. However, how players observe each others' behaviors and how they achieve mutual trust are not fully known. In this study, we researched the growth of a mutual trust protocol in a trading game in a human-based simulation. We designed and implemented web-based multi-player trading game based on the refusible iterative Anti-Max Prisoner's Dilemma game (rAMPD). In the game, each agent's strategy is described by an automaton and periodically modified by human players. We conducted a long-term human-based evolution of mutual trust using this trading game for approximately one month and observed how the agents' automata changed. Analyses of the high-ranking agents' automata and introspective reports by the human players revealed that the mutual trust protocol is achieved by using the initial trade as a signal for mutual recognition.

## Introduction

Intentional reading by an agent is an important topic in the field of artificial life. Learning the other's intention is called the Theory of Mind (ToM), and under the social brain hypothesis, it is thought to be a main factor in the evolution of our brains (Premack & Woodruff, 1978) (Byrne & Whiten, 1989). Being able to estimate the intentions of other people and trust them are important factors in trading in the real world and require intelligence. The "power of trust" becomes larger if an agent's reward is maximized or its penalty is minimized through trading; i.e., trading is encouraged when a winning agent gets a large reward and the losing one incurs only a small loss. Fisher and Shapiro used iterative arm wrestling for teaching the importance of trust in trading (Fisher & Shapiro, 2005). They demonstrated that if two players play an iterative arm wrestling game and the winner gets a reward in each match, it is better for both players to fix the game rather than engage in a real fight. They also showed that the key factors in agreeing to fix a game is that each player needs to be intelligent and trust that after if he or she intentionally loses a match, his or her opponent will intentionally lose the next match. They showed that mutual trust sometimes emerges even without words being exchanged between players.

The Iterative Prisoner's Dilemma (IPD) is a typical game in game theory, and it is designed in such a way that the reward is maximized if both players cooperate (Axelrod, 1984). A cooperative strategy in the IPD is achievable without players having to estimate each other's strategy. This kind of game model is appropriate for simulating ecological behaviors of animals that do not relate to ToM (Le & Boyd, 2007). On the other hand, it is insufficient for representing mutual trust in trading situations because mutual trust requires delayed actions. A human player can lose an arm wrestling match and still believe his or her opponent will lose in the next match.

The current study is on a human-based multi-agent simulation of a refusible iterative Anti-Max Prisoner's Dilemma game (rAMPD). It was conducted to see how mutual trust in trading arises. The Anti-Max Prisoner's Dilemma (AMPD) was first proposed by Angeline (Angeline, 1994). He modified the reward table of IPD so that it could cover the mutual trading behavior of Fisher and Shapiro's iterative arm-wrestling game. We included refusal as the third choice of the agent in AMPD (hence, refusible AMPD, or rAMPD). This extension can simulate real-world trading because each player has the right to ban opponents in free trade. We recruited 74 people to play the rAMPD in a simulation lasting 28 days, and the results of our analyses show how mutual trust arose during this game.

The following sections are organized as follows. Section 2 explains game rule of rAMPD. Section 3 explains how we implemented the system for human-based evolution and conducted experiments and the result of the experiment is shown in section 4. Section 5 analyzes the result and discusses how mutual trust and other strategies are acquired by agents. Section 6 describes how our result contributes to other research field and section 7 describes our method's limitation. Section 8 concludes the paper.

## Game Rules

Table 1 is the reward table of the trading game. The standard IPD conditions are shown in equation 1, and the AMPD conditions are shown in equation 2.

Table 1. Reward table of Trading Game

B \ A	Cooperate	Defect
Cooperate	$(A:c, B:c)$	$(A:a, B:b)$
Defect	$(A:b, B:a)$	$(A:d, B:d)$

$$a > c > d > b, \quad a + b < 2c \quad (1)$$

$$a > c > d > b, \quad a + b > 2c \quad (2)$$

We also added ‘refusal to trade’ as a choice for each agent. If the refusal is selected by an agent, the two agents finish their trade with no chance of retrying. We modified Axelrod’s reward table ( $a = 5, b = 0, c = 3, d = 1$ ) because it is commonly used in game theory simulations. To increase the value of refusal selection, we averaged four constants. We subtracted 3 from  $c$  and subtracted 2 from the remaining three. The reward table for rAMPD was thus ( $a = 3, b = -2, c = 0, d = -1$ ). The average of the four constants was 0, and this satisfied equation 2. The value  $c (=0)$  represents an example of Fisher and Shapiro’s arm wrestling that both player’s cooperative hands do not make sense.

In the rAMPD game, all agents traded with each other iteratively. We also selected the maximum matches in one trade up to 100 times. All agents traded in a round robin fashion. The round robin was repeated several times. The human participants could improve their agent’s strategy between each round robin.

## Human-based Evolution

### Notation of the strategy by automaton

Each participant got his/her own agent and input strategy of the agent through an automaton. We selected automaton-based description of strategy in three reasons. First, automaton-based strategy is understandable to participants especially who are not familiar about programming. Second, the automaton is easy to analyze because of its simple notation. Third, the automaton has enough describable for complex strategy.

Each participant input their agent’s strategy by using a finite state automaton. Each state in the automaton had numbers representing cooperation and defection of the agent. Even states represented cooperation, odd states represented defect, and 0 represented a refusal. The transition arrows between states were described with triplets numbers. The first number represents the present state, the second number represents the opponent’s hand (0 means cooperate and 1 means defect), and the third state represented the next state (even, odd, or 0 state). Each participant described their strategy using the start state number and several triplets. For example,  $\{2\}, \{2,0,2\}, \{2,1,2\}$  means a strategy that is anytime cooperative.  $\{1\}, \{1,0,1\}, \{1,1,0\}$  means coward exploiter. If it is once attacked, it refuses trade.  $\{2\}, \{2,0,2\}, \{2,1,1\}, \{1,0,2\}, \{1,1,1\}$  shows the strategy of tit-for-tat which is famous in IPD. We found that the finite state automaton made it easy for

players to understand each others’ behaviors, and that it is enough descriptive to maintain mutual trust.

### Implementation of the game: how to motivate participants

For motivating participants, we designed the simulation as an online game. All games were implemented in AJAX style, and participants input their strategies using a web form shown in Fig. 1. Each participant could download his/her agent’s trade history from the website at any time (shown in Fig. 1 top). The results of a trade were calculated on the server side and feedback to participants both ranking page and interactive result viewer. Each participant could replay their previous result in viewer mode (shown in Fig. 1 bottom).

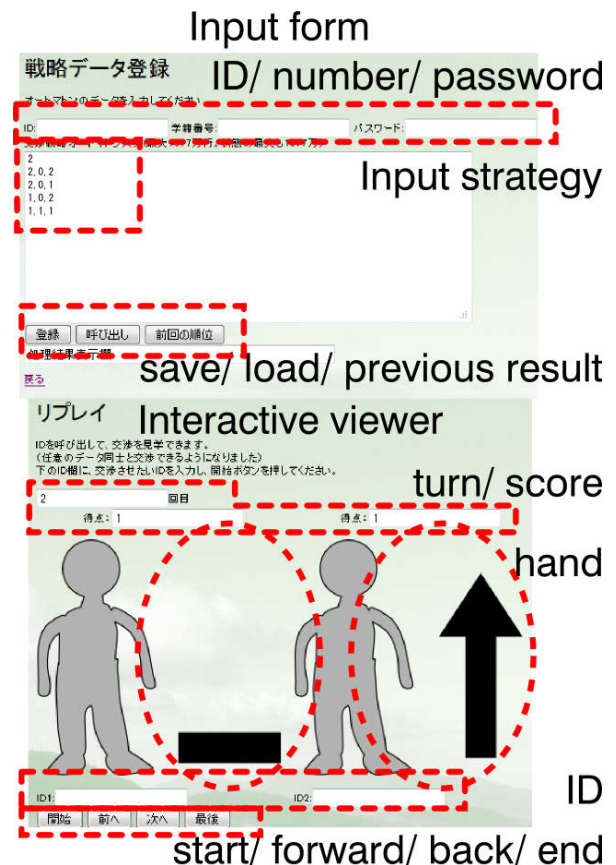


Figure 1. Implemented game screen. Top figure shows the input form of the automation code of the text. Bottom figure shows the viewer mode. Both figures are captured from web browser.

We also wrote a cover story for the game to nurture the imagination of the participants and motivate their play. In the cover story, the participants were residents of an island, and they traded fish in a poisoned pond. Each fish became edible if it had been dipped in a different pond. Each agent could choose between three selections, to wait at home (C), to go to

an opponent's home and take fish (D), or to lock the door (refusal). If both agents wait at home, there was no reward and no penalty (0, 0). If both went the other's home, the doors were locked and both became tired (-1, -1). If one went to the opponent's home and the opponent was waiting there, he or she could eat the fish and the opponent lost it (+3, -2). If one of the players refused, the communication and trading stopped. This story is a bit artificial. However, every participant understands how the rule works.

## Participants

We conducted an experiment in a class learning about automata, and the participants were students from that class. In total, 74 participants played the game. The experimental period ran from 2012/5/29 12:30 to 2012/6/26 8:30. Trading was conducted four times a day during breaks between classes (8:30, 12:30, 16:30, and 20:30), and the ranking table was updated during this period. The chances for updates totaled 112 times. Because updates were done during breaks, each participant had enough time to input strategies and confirm the update's result before and after trading.

All participants were given scores for their class work according to their ranking at the end of the simulation. We divided up the participants who took more than 1 point into 16 groups (these agents were survivors which ate fish and were not hungry). Each group member got from 20 points to 5 points in order to his/her agent's score. The rules were described to the participants before the game started.

It was important for us to confirm that there were no ethical problems in conducting this experiment as a part of the automaton class because it was designed as both an experiment and as a means for students to learning the basic behaviors of automata. The experiment also included an evaluation of the students.

## Results

There were 1109 updates of the automata. The average number of update accesses per trade was 9.9. The average number of updates per player was 15 times. 68 agents achieved more than 1 point at the time of the last update, and their programmers were given bonus points in the class. According to their acquired score, we named each agent in order of highest score A01 to lowest score A74, categorized the 68 participants whose agents exceeded 1 into groups 20 (G20) to 5 (G5), and put 6 participants with less than 0 points in group 0 (G0). The average length of the agents' automata was 33.7. The average length of automata with more than 1 point was 36.5. Figure 2 shows the average length of automaton and average points in each group.

We categorized each player's state using the following rules. If both players each got more than 40 points and less than 60 points, we considered that both players trusted each other and categorized them into the mutual trust (MT) group. If one of the players got more than 50 points and the other got less than -50 points, we considered that one of the players exploited (EX) the other player and that other player was exploited

(EXd). If both players got less than -50 points, we considered that both players could not trust each other and mutual destruction (MD) occurred. If both players got less than 10 points and trading was stopped by one of the players, the trading was banned (BA). If both players got less than 10 points and trading continued until the end of the simulation, trading resulted in stagflation (ST). All categories are shown in Fig. 3. ST only happened in the lower-rank group G0. There are trends on Fig. 3 that higher-ranked agents achieve more mutual-trust than lower-ranked agents (note that BA in high-ranked agents are still required to prevent lower-ranked agents' attacks).

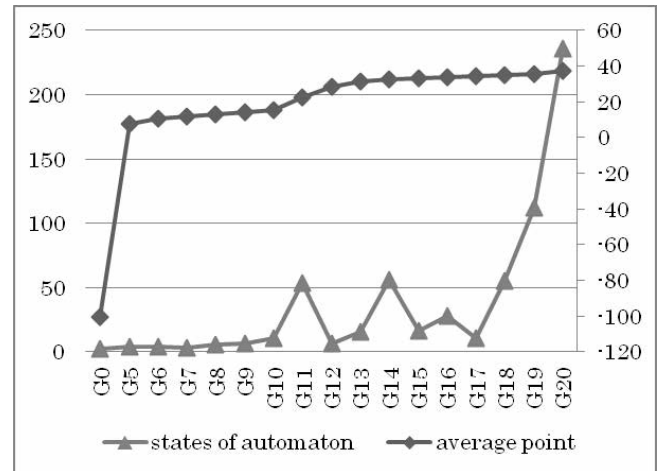


Figure 2. Average states of automaton and average points for each group. The right axis shows the number of automaton states, and the left axis shows the average number of points. The bottom axis shows the 17 groups.

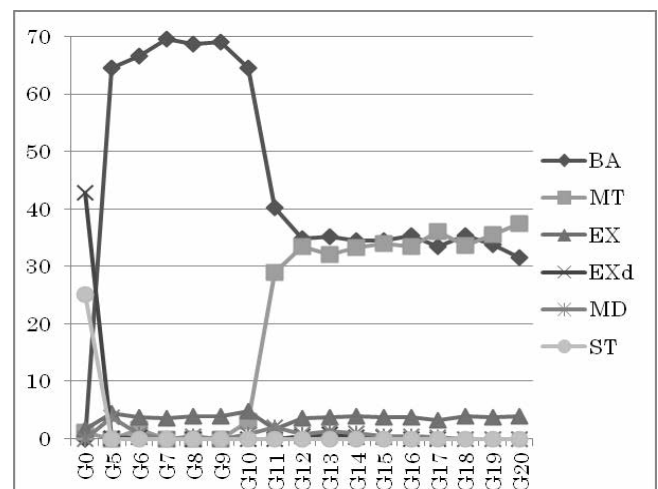


Figure 3. Average number of categories in each group. The bottom axis shows the 17 groups.



## Discussion

### The General strategy of the high-ranked agent

Figure 4 shows a strategy of the high-ranked agent as a meta automaton. All high-ranked agents (in the G20, G19, and G18 groups) had four phases of their strategy. First, the agent repeated cooperate or defect in a determined order. This determined order was different in each agent. If the opponent selected a different hand, the agent transited to the mutual trust phase. In the mutual trust phase, the agent tried to take a complementary hand. If the opponent selected C continuously, the agent transited to the exploiting phase and tried to exploit the opponent. On the other hand, if the agent detected D continuously, the agent transited to the refusal phase and finished the trade

The detailed transition rules depended on the agent. Note that in this game, two identical automata do not succeed because they cannot change to different hands. This restriction discourages users from cheating and accelerates the evolution of the identification process.

From the participants' reports, we confirmed that participants gradually came to understand the several dilemmas in this game. For example, if the identification process is too strict (using lots of confirmation before mutual trust), the opponent may regard that it is impossible to cooperate and simply refuse the trade. This loses the chance of possible cooperation. However, if the identification is too loose, the opponent may think that the agent is too foolish to cooperate and start to exploit it. This also loses the chance to cooperate and reduces rewards.

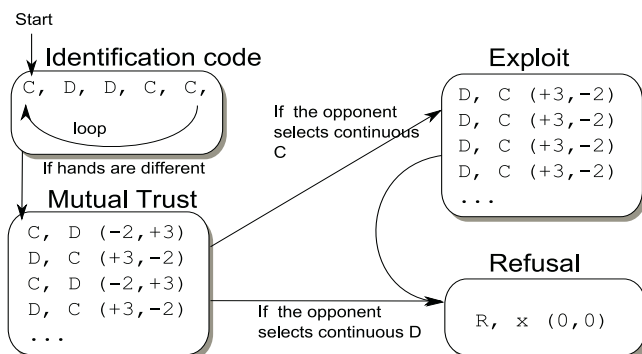


Figure 4. Example of strategy in high-ranked agent (the identification code is extracted from A02)

We studied the results of these three methods by conducting a statistical analysis, manual analysis (where the author input hands manually, traded with each agent, and observed the behaviors), and collecting reports from the participants. The next subsection discusses the analyses conducted during each phase.

### Identification Code Phase

We confirmed that at least 30 agents (A01-A30) had an identification code phase of manual analysis and the participants' reports. Each agent in 30 agents has a different set of hands on the start process (CDDCC, DCDC, etc.). If each agent's hand is different, they start to go mutual trust phase.

The length of the identification code loop was less than five pairs (For example, A02 had 5 loops and A04 had 3 loops). Theoretically, a  $2^5$  bit unique code is required to identify 32 agents. This result corresponds to the fact that the agents that had identification phases numbered less than 30.

57 participants selected D as the first hand of the agent and 17 participants selected C. Most participants selected D because the agent had a chance to get a higher score than the opponent. On the other hand, the participants who selected C reported that selecting C in the first hand had an advantage because it was easier to start up a mutual trust situation with it than D. One of the participants reported that his strategy followed an old proverb "win by losing."

### Mutual trust phase

Theoretically, mutual trust arises in a longer loop, like CCDD, CCCDD, etc. However, all agents used short mutual trust loops (CDCDCD...). 42 agents (A01-A41 and A61) had mutual trust phases. As shown in Fig. 3, high-rank agents had more mutual trust (MT) and fewer refusals (BA). This result suggests that lower-rank agents lost the chance through their own or their opponent's (BA) refusal, whereas high-ranked agents could use the chance to make a mutual trust loop (MT). A larger set of states in an automaton weakly suggests that mutual trust requires each automaton to have more complex states. This result suggests the validity of the social brain hypothesis wherein evolution of intelligence (approximated by the number of states) is accelerated through identification in society (Byrne & Whiten, 1989).

### Exploiting Phase

The exploiting phase was observed in almost all agents (A01-A71). There were 4 exploited agents (A71-74, all agents were in G0). Continuous Cs was a trigger for transiting to the exploiting phase. We confirmed that all agents who had an exploiting phase transited to the exploiting phase after 2 to 4 continuous Cs. As shown in Fig. 3, almost all agents attacked during the transit from the exploiting phase to the refusal phase.

Keeping the exploiting phase has a risk in that it may imprudently trigger the opponent's refusal phase. However, the exploiting phase was preserved until the end of the game because there are big rewards for exploiting phase. There were three participants who input only C regardless of their opponent's hand and did not change strategies (A72-A74). A71 had a simple strategy that selected a CCDDCCD... loop unrelated for the opponents' hands. These weak agents kept being exploited.

## Refusal Phase

The refusal phase was inevitable because if the opponent selected continuous D regardless of its hand, the agent prevented a negative score just by selecting refusal (C,D and D,D are both negative). 41 agents had a refusal phase (A01-A41).

All of these agents reacted to more than 3 continuous Ds. On the other hand, several agents allowed 1 or 2 Ds. These behaviors kept the opponent cooperative and not "be anger" (for triggering opponent's refusal attitude).

## Difference between rAMPD and AMPD

Each agent acquired more complex strategy in rAMPD compared with a strategy in AMPD game proposed by Angeline (Angeline, 1994). There are three factors for generating different results from previous research.

The first factor is difference of simulation. Angeline uses computer-based simulation for evolving each agent's strategy. On the contrary, we used human-based simulation. The latter condition expands the possibility of each agent's strategy. The second factor is the description of the strategy. Angeline uses set of four hands (CCCC-DDDD) for describing strategy of each agent. On the other hand, we used automaton for describing strategy. The automaton makes it possible to use more complex strategy. The third factor is the refusal hand of each agent. Refusal phase creates "point of no return" in each trade and it makes communication complex. In Angeline's AMPD game, each agent has no refusal selection and a trade continues to determined cycle. If an opponent plays continue Ds (which means two or more continued D hand), most appropriate strategy is replying with continuous D. If the opponent stops continuous D, the agent just needs to stop continuous D. In our situation, a most appropriate hand for the opponent's continuous D is just refusal. However, if continuous D are produced mistakenly by the opponent, there is still a chance for creating mutual trust in each other. We hypothesized that the refusal phase is the critical factor for evolving complex communication between agents. As a future work, we will confirm the hypothesis by using computer-based simulation.

## Contributions

We tried a human-based multi-agent simulation instead of a computer-based one. The human-guided approach is used in several fields, from artificial life, cloud sourcing, and human interfaces (Kosorukoff, n.d.)(Paolacci et al., 2010)(Osawa & Imai, 2012). Our results suggest that this approach also works if the motivations of the human players are carefully designed.

Our findings revealed two important factors related to game theory and multi-agent simulation. The first is in regard to the emergence of mutual trust in trading itself. In game theory, the possibility of mutual trading can be analyzed in the Cheap Talk Game that divides a trading game into an initialization phase and a main trading phase (Wärneryd, 1991). Our results suggested that identification of others and mutual cooperation can be achieved even without "cheap talk" by using the reward

itself. This finding may lead to multi-agent simulations becoming simpler as far as their requirements go. The second factor is the importance of being able to refuse during free trade. Previous studies mainly focused on the locality of the agent as a way of avoiding agents they were not confident about and this leads to agents forming clusters (Suzuki & Arita, 2001). This approach is good for ecological simulations. However, general free trade is not dependent on the distance to the others, but rather on the mutual intention of trading. Our results suggest that agents come to believe each other and reject agents they are not confident about not by using additional information (like cheap talk and location) but rather through behaviors.

In light of the above discussion, we think that our human-based multi-agent simulation of the rAMPD game reflected the essence of real-world free trading and gave us good insights about how identification of others and mutual trust arises in humans.

Last, we want to note that human-based multi-agent simulation quickly proceeds analysis for the game space because we can collect agent's process of evolution by participants' introspections. We want to emphasize that most participants are motivated by this gamification method (students involved in our "homework" make good scores in class). We believe that motivated participants are very good research factors for estimating the possibility of game space especially in earlier stages of study.

## Limitations

Human-based simulations are dependent on capricious humans. To handle human resources properly, we need to design the experimental setup carefully.

Although there were 112 updates in this task, almost all of them happened during the first week and final week. To maintain motivation during the whole game period, we may need to back-reward the participants (for example, by scoring on a weekly basis).

Three participants did not update their agents, and this sabotage influenced the other participants. Several participants complained in the report that the authors did not evict these three agents. We think that these variable motivations also reflected a real simulation. However, this result also shows the importance of a good motivation design in human-based multi-agent simulation. This underscores the need to carefully design the agent's goal - each participant's motivation - in a human-based multi-agent simulation.

Knowing the number of trades may increase the unwanted factors. The top scoring groups (G20 and G19) had more than 100 states in their automata. The analyses of the automata and the reports from the participants showed that a large number of states were prepared for the 100th match. Defect or refusal is the optimal strategy even if mutual trust arises because the 100th match does not have a succeeding match. There were also three agents that prepared the 100th match in G11, G14 and G16, as the spikes in Fig. 1 show. The reward (<5 points) for defect or refusal in the 100th match was relatively small

compared with the points from MT (around 50 points) and EX (around 300 points). The main ranking seemed dependent on the amount of mutual trust and the 100th match did not influence mutual trust. The increasing trend in MT in going from G10 to G20 (Fig. 2) supports this idea. These unwanted evolutions can be avoided if the number of matches is indefinite in each update.

Human-based simulations sometimes encounter ethical problems. For example, we could not regulate communications between participants, unlike in the case of a computer based multi-agent simulation. In this experiment, the participants were rivals and there was no real motivation to cooperate. Moreover, cloning was meaningless in this task. These two facts restricted communication between participants. The participant reports also suggested that there was no cooperation between the participants. However, it is hard to monitor the sorts of strategy that could have been generated through discussions with other participants. This problem may be avoided if the game is conducted online anonymously and all behaviors are monitored. Anyway, the experimenter must be careful about regulating human behaviors. It is important to ensure that the experiment is profitable for the participants themselves.

## Conclusion

We designed and implemented a web-based multi-player trading game based on the refutable iterative Anti-Max Prisoner's Dilemma game (rAMPD). In this game, each agent's strategy is described by an automaton and is periodically modified by human players. We conducted a one-month human-based multi-agent simulation using this trading game lasting approximately one month and observed how the agents' automata changed. Analyses of high-ranking agents' automata and introspective reports from the human players revealed that a mutual trust protocol arises using the initial trade as a signal for mutual recognition.

## Acknowledgements

Our work was supported by the diligent students at Keio University. This work was supported by the JST PRESTO program.

## References

- Angeline, P. J. (1994). An Alternate Interpretation of the Iterated Prisoner's Dilemma and the Evolution of Non-Mutual Cooperation. *Proceedings of 4th artificial life conference* (pp. 353–358).
- Axelrod, R. (1984). *The Evolution of Cooperation*. Basic Books.
- Byrne, R. W., & Whiten, A. (1989). *Machiavellian Intelligence: Social Expertise and the Evolution of Intellect in Monkeys, Apes, and Humans*. Oxford University Press, USA.
- Crawford, V. P., & Sobel, J. (1982). Strategic Information Transmission. *Econometrica*, 50(6), 1431 – 1451.
- Fisher, R., & Shapiro, D. (2005). *Beyond Reason: Using Emotions as You Negotiate* (p. 256). Viking Adult.
- Kosorukoff, A. (n.d.). Human based genetic algorithm. *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236)* (Vol. 5, pp. 3464–3469). IEEE. doi:10.1109/ICSMC.2001.972056
- Le, S., & Boyd, R. (2007). Evolutionary dynamics of the continuous iterated prisoner's dilemma. *Journal of theoretical biology*, 245(2), 258–67. doi:10.1016/j.jtbi.2006.09.016
- Nowak, M. A., & May, R. M. (1992). Evolutionary games and spatial chaos. *Nature*, 359(6398), 826–829. doi:10.1038/359826a0
- Osawa, H., & Imai, M. (2012). Possessed Robot : How to Find Original Nonverbal Communication Style in Human-Robot Interaction. *International Conference on Agents and Artificial Intelligence (ICAART)* (pp. 632–641). INSTICC.
- Paolacci, G., Chandler, J., & Ipeirotis, P. G. (2010). Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5(5), 411–419. doi:10.2139/ssrn.1626226
- Premack, D., & Woodruff, G. (1978). Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(04), 515–526. doi:10.1017/S0140525X00076512
- Suzuki, R., & Arita, T. (2001). Evolutionary Analysis on Spatial Locality in the N-person Iterated Prisoner's Dilemma. *Proceedings of inaugural workshop of artificial life*, 105 – 114.
- Wärneryd, K. (1991). Evolutionary stability in unanimity games with cheap talk. *Economics Letters*, 36(4), 375–378. doi:10.1016/0165-1765(91)90201-U

# Hebbian Learning In A Multimodal Environment

Julien Hubert, Eiko Matsuda and Takashi Ikegami

The University of Tokyo, Ikegami Laboratory, Tokyo, Japan  
 {jhubert,eiko,ikeg}@sacral.c.u-tokyo.ac.jp

## Abstract

Hebbian learning is a classical non-supervised learning algorithm used in neural networks. Its particularity is to transcribe the correlations between couple of neurons within their connecting synapse. From this idea, we created a robotic task where 2 sensory modalities indicate the same target in order to find out if a neural network equipped with Hebbian learning could naturally exploit the relation between those modalities. Another question we explored is the difference in terms of learning between a feedforward neural network(FNN) and spiking neural network(SNN). Our results indicate that a FNN can partially exploit the relation between the modalities and the task when receiving a feedback from a teacher. We also found out that a SNN could not complete the task because of the nature of the Hebbian learning modeled.

## Introduction

One important aspect of our everyday life is our capacity to acquire knowledge by experiencing our environment. Animals possessing the capacity to learn can detect and exploit the correlations present in their environment. This gives them, among other advantages, the capacity to predict their world and avoid undesirable outcomes, which is akin to Friston's free energy theory where living systems try to minimize their free-energy in order to increase their capacity of prediction of their environment(Friston (2010)).

The neurobiology of learning in the brain is driven by synaptic plasticity, also referred to as Hebbian plasticity or Hebbian learning after Donald Hebb who first proposed a theory of how learning could take place(Hebb (1949)). The general idea drawn from Hebbian learning states that when 2 neurons are connected together by a synapse, the correlated activity between the two would evoke structural modifications within the synapse such the capacity of the pre-synaptic neuron to cause potentiation in the post-synaptic one would increase(see Abbott and Nelson (2000) for a complete introduction). This phenomenon was observed at the neuronal level by Kelso et al. (1986).

The task we are interested in is a multi sensory integration task where a robot must reach a target indicated by a sound and a light source. The main question we want to address

is if a neural network equipped with Hebbian learning can naturally extract and exploit the correlation between the two sources. Indeed, sound and light are two sensory modalities with different properties and different yield. It would then be interesting to know if Hebbian learning has the potential of integrating them transparently, that is without any external mechanism for this task. It is generally expected that Hebbian learning can detect correlations at the neuronal level, but our question is if it can also integrate the correlations present in the environment in order to convey additional properties or abilities to its host. For instance, in this particular task, we are interested in the robustness that Hebbian learning can provide when noise is added on the sensors. Another aspect we explore is the effects of asynchronous activation of the sensory modules on the behaviour of the robot. The particularity of our experiment is that no mechanism helping the integration of the two sensory modalities is provided, and any emergent property of the system can only be the result of the Hebbian learning.

As a side experiment, we were also interested in the comparison between classical rate based neural networks and spiking based neural networks when it comes to Hebbian learning. Spiking neural networks(SNN) are a relatively recent paradigm where information transferred between neurons is no longer a continuous value, but a temporally limited event representing a neuron discharging its membrane potential (Maass (1997)). SNNs represent a model closer to the neurobiology of the brain, and as such possess also a model of Hebbian learning directly drawn from neurophysiological recordings. The particularity of this learning is that it incorporates a window of time where a synapse can be modified, which differs from the Hebbian learning used in classical NN where the changes are instantaneous. Our question is whether this architecture would lead to different behaviors. We will see that SNNs could not reproduce the behavior obtained by the traditional neural network, and we will discuss the causes and the impact of this result.

The article is divided in 3 sections. The first presents the experimental setup and includes a description of the task and of the controllers. In the second section are presented the



results for both controllers which are discussed in the last section.

## Experimental Setup

### The Task

Our experiments use a robot whose task is to move toward a target area in its environment. The robot moves in an open environment where one light source and one sound source are located at the same position. Facing those sources is a grid of 7x7 cells (see figure 1). Those are used as starting positions for the robot. The task of the robot is to navigate its environment using its sensors until it reaches a point at a maximum distance of 1 grid cell from the sources. The robot has 6 minutes to complete this task. If the robot reaches the goal within that time period, the trial is considered a success; otherwise, the trial is counted as a failure.

The robot and its environment are simulated in all the experiments. The simulation of the robot is based on the e-puck robot equipped with light and sound sensors (Mondada et al. (2009)). The simulation of the environment is based on data obtained from a real robot in a real environment. A model of the light and of the sound has been built using sensor readings gathered from a robot moving in the environment. This has consequences on the simulated sensors. The simulated light sensor can not perceive the emitted light from the source outside of its field of view. The simulated sound sensor does not only perceive the sound from the source, but also the sound from the motors of the robot. The noise generated by the motors can be louder than the emitted sound, preventing the robot from perceiving it. The real source of the light is a white neon bulb while the real sound source emits white noise. Figures 2 and 3 show respectively the light and the sound perceived by the robot at the different positions in the grid, the goal being located at coordinates (3.5, 7) on the graphs, i.e. at the center column of the top row. The frequency of update of the simulation depends on the neural network. In the case of the feedforward neural network, a timestep of 0.05s is used. For the spiking neural network, a timestep of 0.001s is used.

The performance of the robot is measured by its capacity to reach the goal in less than 6 minutes. For a single trial, the performance of a robot would be one if it reached the goal and zero otherwise. The duration of the trial is not considered in the performance measure.

### The Controller

As mentioned in the introduction, two controllers are tested in this task. The first one is a standard feedforward neural network (FNN). The second is a spiking neural network (SNN). Both are equipped with plastic synapses following a Hebbian rule. The number of input, hidden and output neurons remains the same with both controllers. The next subsection describes the implementation using the feedforward network. The differences with the spiking neural

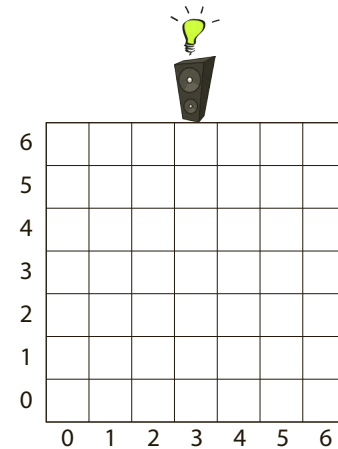


Figure 1: Experimental arena. Each cell of the grid is a possible starting position for the robot.

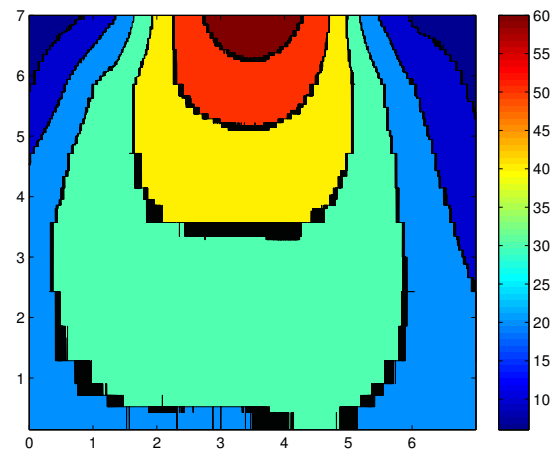


Figure 2: Robot's perception of the light.

network will be explained later on. The next section will detail the FNN.

**Feedforward Neural Network** The FNN possesses 7 inputs and 4 outputs. No hidden neurons are present. The input and output layers are fully interconnected. The weights are tuned using Hebbian learning (Hebb (1949)). A visualization of the network is shown in figure 4.

The inputs of the FNN are divided into two groups. The first group is composed of 4 inputs and handles the perception of the sound. The second group contains 3 inputs and receives the perception of the light. In both cases, the inputs are not the outputs of the sensors, but a pre-processed version. Each sensor is connected to a memory containing the past readings of each sensor. In the case of the sound, the size of the memory is 30 readings. For the light, it is only 10 readings. The difference in memory size can be ex-

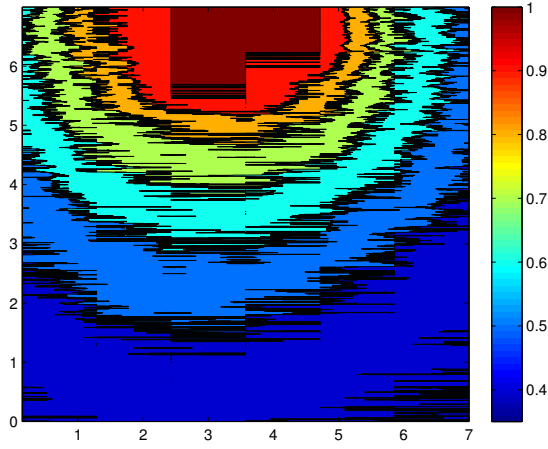


Figure 3: Robot's perception of the sound.

plained by the difference in noise between the two sources. The data from the sound sensor shows a higher level of noise while the light sensor shows a more gradual increase. This difference will become important when explaining how the inputs are computed. The light sensor also possess a single value memory updated every 120 timesteps and containing a single reading. We will refer to it as *imprint* later on. This imprint serves as an ambient light measure used by the controller to discriminate the light coming from the source.

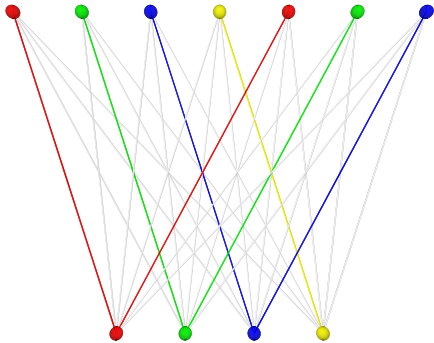


Figure 4: Network composed of 7 input neurons(top) and 4 output neurons(bottom). The lines represent the synapses. The colors used represent the ideal connections between nodes. For instance, both red inputs should be ideally connected to the red output.

The controller described below has been first designed on a real robot and then transposed to simulation. Because the noise in a real environment differs from a simulated one, this approach guarantees that the behavior in simulation remains

the same as in reality.

**Inputs Pre-Processing** The inputs of the FNN are pre-processed in order to obtain binary inputs. The pre-processing is necessary for the Hebbian learning to be stable and is different for each type of sensor. Only one input can be activated for each sensor at every timestep. Practically this means that among the 7 inputs, 1 input from the first 4 and 1 input from the last 3 can be activated simultaneously.

The following equations describe the activation rule for the first 4 inputs relating to the sound perception:

$$I_0 = 1 \quad \text{if} \quad S(t) > S(t - 30) + 0.03 \quad (1)$$

$$I_1 = 1 \quad \text{if} \quad S(t) < S(t - 30) + 0.03 \quad (2)$$

$$I_2 = 1 \quad \text{if} \quad S(t) \geq S(t - 30) \quad (3)$$

$$I_3 = 1 \quad \text{if} \quad S(t) \leq S(t - 30) \quad (4)$$

where  $I$  are the inputs and  $S(t)$  refers to the sound sensor reading at time  $t$ . Those equations are evaluated in order and the evaluation stops when one input is set to 1.

The inputs relating to the light sensor follow a set of rules also. Before deciding which input to activate, the imprint must be subtracted from every reading used. The following equations describe the activation rule the last 3 inputs relating to the light perception:

$$I_6 = 1 \quad \text{if} \quad L(t) < 0.01 \quad (5)$$

$$I_5 = 1 \quad \text{if} \quad \begin{cases} L(t) < L(t - 10) - 0.01 \\ \text{the robot goes backward} \end{cases} \quad (6)$$

$$I_6 = 1 \quad \text{if} \quad \begin{cases} L(t) < L(t - 10) - 0.01 \\ \text{the robot goes forward} \end{cases} \quad (7)$$

$$I_5 = 1 \quad \text{if} \quad \text{if the robot goes backward} \quad (8)$$

$$I_4 = 1 \quad \text{if} \quad \text{if the robot goes forward} \quad (9)$$

where  $I$  are the inputs and  $L(t)$  refers to the light sensor reading at time  $t$ . As for the sound, those equations are evaluated in order and the evaluation stops when one input is set to 1.

The equations for the sound and the light sensors differ because of the properties of the sensors. For the sound, equations 1 and 2 target cases when the current level of sound differs from the past by more than a threshold. Equations 3 and 4 are used when the current level is below this threshold. For the light, equation 5 is applied when the light has no significant differences from the ambient light. Equations 6 and 7 deals with the case where the current level of light is lower than 10 readings ago. Finally equations 8 and 9 are used when there is no important change in the current light level. For the light, the direction of movement of the robot had to be taken into account as the sensor is directional.

The parameters used in the above equations were determined experimentally for our setup.

**Outputs** The outputs are squashed to a range of  $[0; 1]$  using the sigmoid function and are used to activate specific behaviors. Each output is attached to one behavior through a winner-takes-all strategy, i.e. the output with the highest activation activates its corresponding behavior. The four behaviors are:

1. Output 0 maintains the current behavior.
2. Output 1 inverts the current behavior. If the robot is going forward, it will go backward at the next timestep and vice-versa.
3. Output 2 modifies the current behavior to create a left turn while maintaining the same direction.
4. Output 3 modifies the current behavior to create a right turn while inverting the current direction.

**Learning** As mentioned previously, the weights of the FNN are tuned through learning using Oja's Hebbian rule (Oja (1982)). The rationale behind this choice is to allow the possibility for different couplings between the different sensory modalities. Despite the unsupervised nature of Hebbian learning, we cannot expect that the FNN will learn the task without supervision. For that purpose, we implemented a crude controller that will indicate which output should be activated based on the activated inputs. Once the FNN's outputs have been read and a behavior activated, those are replaced by the outputs advised by the teacher. It is possible that 2 outputs be activated at the same time as the light inputs does not necessarily indicate the same behavior as the sound inputs. In that case, two outputs are activated with a strength of 0.5. If both set of inputs concur on one behavior, its assigned output will receive an activation of 1. This will express the certainty of the decision of the teacher and influence the learning. Once the outputs have been modified, the Hebbian rule is applied and the weights are modified. The taught connections between inputs and outputs are as follows:

- Output 0 should be connected to input 0 and input 4
- Output 1 should be connected to input 1 and input 5
- Output 2 should be connected to input 2 and input 6
- Output 3 should be connected to input 3

Those ideal connections are also shown in figure 4, where inputs and outputs sharing similar colors should be ideally connected by the synapse of the same color.

**Spiking Neural Network** The SNN is based on Izhikevich neurons (Izhikevich (2003, 2004)). Contrary to feedforward neurons who possess a membrane potential that is continuous and transferred directly to other neurons, a spiking neuron transfers information in the form of spike. A spike

is an electrical impulse sent by a pre-synaptic neuron to all its post-synaptic neurons. The synaptic plasticity follows the model of STDP proposed by Song et al. (2000). This model differs from Oja's Hebbian rule used for the FNN as it incorporates a time window during which 2 spiking events will produce a synaptic modification. In other words, the firing of the pre- and post-synaptic neuron does not have to be simultaneous to produce a synaptic modification. The temporal sequence of firing is nevertheless important. If the pre-synaptic neuron fires before the post-synaptic one, the synaptic strength is reinforced. In the opposite case, the synapse is weakened.

The Izhikevich neurons are determined by 4 parameters. The values we chose to use correspond to the regular spiking model ( $a = 0.02$ ,  $b = 0.2$ ,  $c = -65mV$  and  $d = 6$ ). The parameters for the STDP are  $A_+ = 0.02$ ,  $A_- = 0.021$ ,  $\tau_+ = \tau_- = 0.02$ . For the SNN to be mathematically stable, the time step of the simulation has been increased to 0.001s. All the timings of the simulation have been adapted to reflect this change and to provide the same setup with the SNN than with the FNN.

## Results

### Feed Forward Controller

Using the FNN as controller, the robot manages to reach the target from all positions within the grid. The average success rate from every starting point in the grid, over 1000 repetitions, is shown in figure 5, where the target is located on the middle top position in the arena. The 3 graphs show the success for 3 different conditions. In the top left graph, the light sensor is the only one activated which gives us an idea of its usefulness to reach the target. It is interesting to notice that our robot appears to be better at picking the light coming from its right, as the performance is higher on the left side of the arena. The top right graph shows the success when only the sound sensor is activated. We can see that the performance is not symmetrical but is much more regular over the whole arena than for the light sensor. Nevertheless, the overall performance is lower. Finally, the bottom graph shows the performance when both sensors are activated. The pattern is similar to when only the sound sensor is activated, but we can notice a faster decrease in performance when far away from the target. We can also notice a slight tendency to have a higher performance when the robot is starting from the left side of the arena, which is consistent with what we see when only the light sensors is activated. As such, it is clear that both sensors influence the performance of the robot, but it is also clear that it is not always a positive reinforcement. It seems that both sensors interfere with each other at the level of the controller.

To explore how the sensors are combined by the controller, we performed an experiment where light processing and sound processing can possess different timescales, i.e. each one can be activated independently and at different tim-

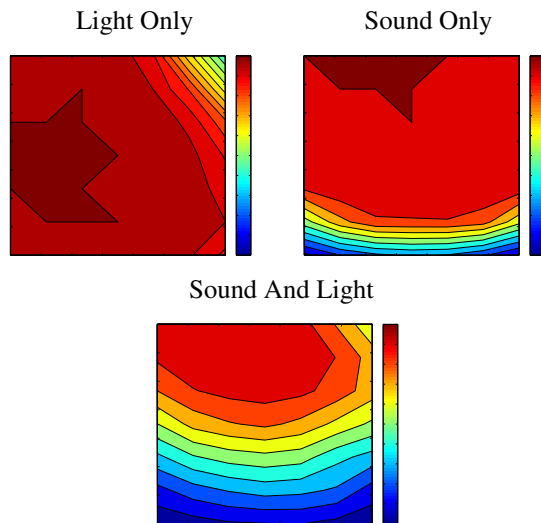


Figure 5: Rate of success for every starting cell within the arena, under 3 conditions: light sensor activated, sound sensor activated, both sensors activated. The goal is located on top central point.

ings. Also, we explored how sensory noise can influence the performance of the robot and if the plastic controller could compensate one sensor by the other. For all experiments, the level of noise describes the max value drawn from a uniform distribution that can be added to a reading of a sensor. The results are shown in figure 6. The left row shows the performance when noise is added to the sound sensor, while the right shows the same with the light sensor. Each row shows a different level of noise for their respective sensor. From top to bottom, the level of noise is increased by increment of 0.2. The X and Y axis provide the speed of update of the sensors. 1 means that it is updated every time step, while 20 means every 20 time steps. The X-axis provides the update speed of the light sensor, while the Y-axis the speed for the sound sensor. The Z-axis is the success rate.

When noise is applied, we notice that the performance is the same for any update speed, except when both modules function at the same speed. In that particular situation, the performance drops slightly, creating a ridge, which could be the consequence of an interference between the two sensory modules. When progressively adding noise, a drop in performance on one side of the ridge appears, but the other side is relatively not affected. The drop on one side only is the consequence of the difference in update speed of the 2 modules. When the noisy module is updated faster than the non-noisy one, this module drives the behavior of the robot more frequently. This is due to the fact that the commands of the noisy module will overwrite the decisions of the other one, as it is activated more frequently. As such, the robot makes more errors as it relies mainly on less reliable information.

When the faster module is the non-noisy one, it can compensate for the errors of the other module, reducing the impact of the noise on the system. We can nevertheless notice that in the area surrounding the ridge, the performance is higher even if the noisy module is updated slightly faster than the other one, which means that both modules are nonetheless interacting and improving the performance of the robot under noise.

Our last test is about the importance of each sensory module in the decision process. As the leading behavior is based on a winner-takes-all strategy at the level of the outputs, it is possible to compute which module takes the decision by observing which input produced the activation in the winning output. Figure 7 shows the rate of dominance of the light module, of the sound module, and the rate of cooperation between the 2 modules for different levels of noise applied on the sound sensor. The graphs show for any cell in the grid the percentage that the light or sound module decides the behavior for that particular cell. The cooperation rate is the percentage of time where the two modules agree.

We can see that the light module is the most dominant over all cells. The sound module has a maximum rate of decision of maybe 20%, while the light module can go up to 80%. This can be expected as the light is the most reliable sensor of the two. The most interesting aspect is how the rates change with the addition of noise on the sound sensor. From 0 to 0.5, we can see that the light module is now dominant over the all arena. This means that the sound being unreliable, it is barely used to drive the behavior. There is also almost no cooperation between the two modules. The sound module becomes relevant only when the robot approaches the sound source, probably because the light sensor at this position becomes saturated and a noisy sound sensor provides more information.

Additionally, we also plotted the average success rate for each condition with a varying level of noise, which can be seen on figure 8. As can be expected, when the light module is only active, the performance remains the same as it is not applied noise to. One visible aspect is that the performance with the sound module slightly increases initially, to decrease after a pick in terms of noise. This is due to the controller having been initially implemented in a real robot where the sound sensors delivered more noise readings. Consequently, our simulated controller performs better under a weak level of noise, which allows it to escape local minima within the sound landscape. Also, we can see that the performance of the sound only condition is initially higher than for the sound & light condition. But this changes above a noise level of approximately 0.2. This implies that the controller can support higher noise levels when both modules are activated.



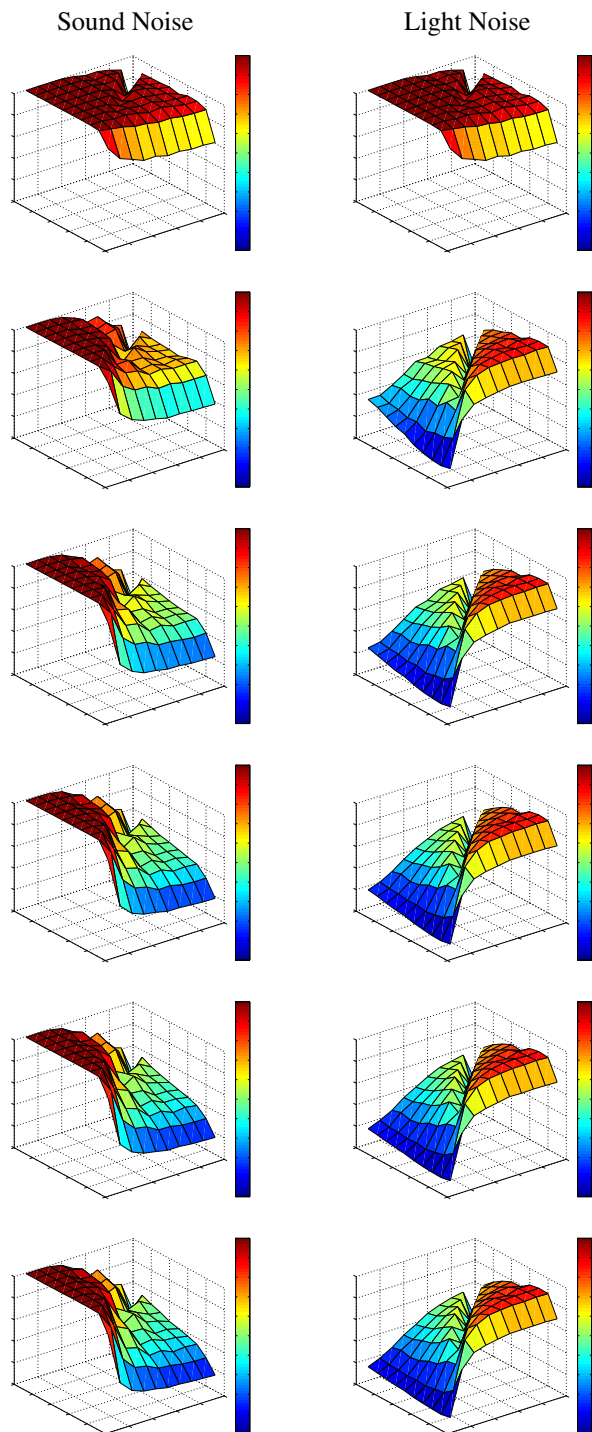


Figure 6: Variation of the performance of the robot under different timescales and different level of noise for the sensory modules. From top to bottom, the level of uniform noise is varied from 0 to 1 by increment of 0.2. The left column has noise on the sound sensor only, while the right one has noise on the light sensor only.

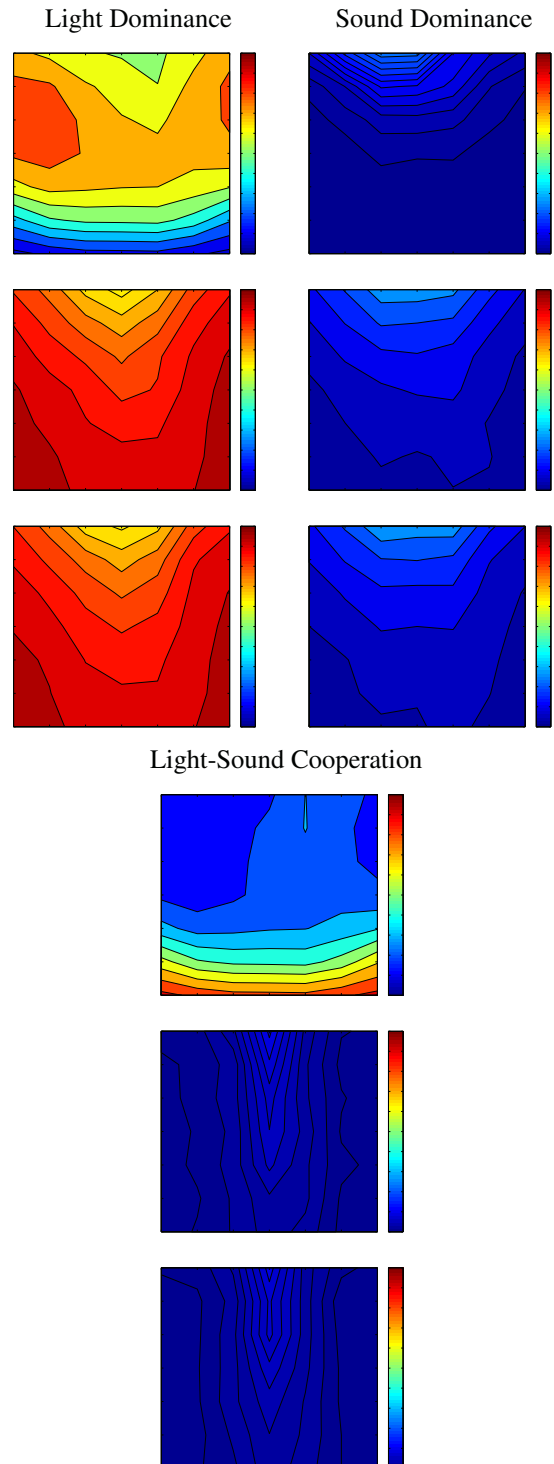


Figure 7: Rate of dominance and of cooperation between the light and sound modules for 3 levels of noise applied on the sound sensor: 0, 0.5 and 1.0.

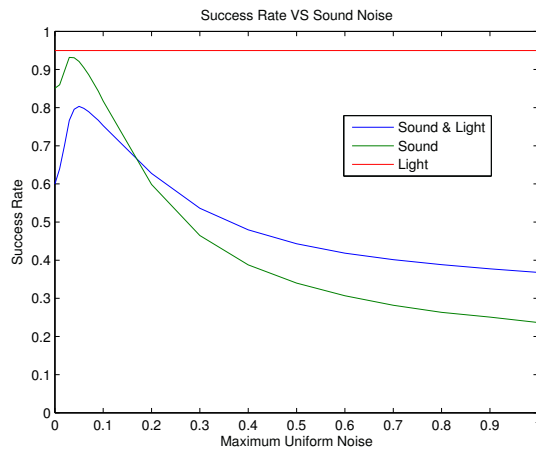


Figure 8: Comparison between the performance of each condition (light only, sound only and both) with different levels of noise on the sound module.

### Spiking Controller

The results with the FFN were interesting so we expected a lot from the SNN. Unfortunately, despite many attempts, we were unable to reproduce the results obtained earlier. The SNN was not capable of extracting the right connectivity from the correlations present in the environment. Fortunately, the reason is quite clear and can shed some lights on the differences between using a FFN and a SNN, and can help orient the design of robotic experiments using a spiking controller.

The reason for the lack of replication of the task can be found in the window of time during which synapses are modified under STDP. This window is roughly 40ms where a synaptic strength can be either reduced or increased. In our case, the problem comes from alternating behaviors. Following what we described previously, input 0 would ideally be connected with output 0, and input 1 to output 1. As the inputs are never activated simultaneously, and the learning chooses a winner-take-all strategy on the outputs, the FFN obtains the right connection as the computation of the change in synaptic strength has no memory of the past activation of the neurons. In the case of STDP, such a memory exists. If two behaviors alternate frequently, and are separated by less than 40 ms, they will interfere with each other and create synaptic connections that should not have been learned. For instance, if input 0 spikes, then we would want output 0 to spike also later on. If this is the case, their connecting synapse is strengthened. If soon after, input 1 spikes, followed by output 1. Then, not only their connecting synapse will be strengthened, but also the synapse connecting input 0 and output 1. Over time, the activation of input 0 will create an activation in output 0 and output 1. This is equivalent to learning the wrong correlations from the environment.

At this point we need to mention that a SNN can of course resolve this task, but it would need a different architecture as the one we offer here. It would need one that can cope with the interferences. Our search for different parameters for the SNN, or by changing some details of the task, did not lead to a successful learning. Our next step should be to research what kind of topology can cope with the interferences. This might imply introducing some kind of modularity in the system.

### Conclusion

At the beginning of the article, we wanted to explore two questions. The first was if a neural network equipped with Hebbian learning could naturally learn and exploit the correlations present in an environment where a target is indicated by two sensory modalities. The second was if there was a difference between a feed forward neural network and a spiking neural network, both equipped with their version of Hebbian learning, in terms of performance on the task. We can start answering those two questions.

Concerning the first question, it is difficult to say that Hebbian learning was sufficient to learn the correlations in the environment. Figure 5 and 8 show that the controller displays a different behavior when the light and sound modalities are provided, compared to the situation when only one of them is used. This tells us that the network is exploiting both sensory modalities, and that the combination is not a simple linear sum of both modules. Furthermore, the dominance analyses shown in figure 7 confirm that based on the amount of noise on one modality, the less noisy one becomes more important in deciding which behavior to activate. This supports the idea that Hebbian learning, combined with a teacher, could naturally exploit the most reliable information present in the environment. The teacher is important because he guarantees that the most reliable connections between inputs and outputs are more frequently reinforced compared to the more noisy ones. Following that idea, it becomes also plausible that Hebbian learning could modify the connectivity of the network to promote one sensory information if it provides the most reliable information within a specific area in the robot's environment. Without the teacher's feedback, it seems difficult for the Hebbian learning to naturally exploit this information. Indeed, deactivating it does not lead to a successful behavior. But through the teacher's interaction, the system managed to rely on the most reliable sensor to complete the task.

The answer to the second question is that there are clearly some differences between Hebbian learning in FFN and SNN. The interferences between the different behaviors prevent the network to learn the relevant correlations from the environment. This clearly demonstrates that there are important differences between the classical Hebbian learning found in FFN and the STDP found in SNN. The time window used by STDP to compute the synaptic modifications

are responsible for the failure to learn our particular task. As this mechanism has been modeled directly from real synapses going through strengthening or depression, ignoring this difference might lead to undesirable effects. For instance, the conclusions drawn from a robotic experiment using FFN and Hebbian learning might not be related to what could be happening in a real neural network because time has not been taken into consideration. As one goal of artificial life is to understand life as it is by recreating it, it might be wise to work with closer models of the brain in order to draw conclusions relevant to what we are interested in.

## References

- Abbott, L. F. and Nelson, S. B. (2000). Synaptic plasticity: taming the beast. *Nature Neuroscience*, 3:1178 – 1183.
- Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11:127–138.
- Hebb, D. O. (1949). *The Organization of Behavior*. Wiley, New York.
- Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572.
- Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, 15(5):1063–1070.
- Kelso, S. R., Ganong, A. H., and Brown, T. H. (1986). Hebbian synapses in hippocampus. *Proceedings of the National Academy of Science USA*, 83(14):5326–5330.
- Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, volume 1, pages 59–65.
- Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273.
- Song, S., Miller, K. D., and Abbott, L. F. (2000). Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3:919–926.

# Follow the Leader: a Scalable Approach for Realistic Group Behavior of Roaming NPCs in MMO Games

Dario Maggiorini, Laura Anna Ripamonti, Samuele Panzeri

Department of Computer Science, University of Milano, Italy  
 ripamonti@di.unimi.it

## Abstract

In modern Massively Multiplayer Online Games, Non Playing Characters (NPCs) moving on the battlefield play a key role in term of user experience: many time players are required – alone or in groups – to fight or avoid them in order to progress in experience. Unfortunately, standard NPCs behavior, i.e., patrolling between rally points, does not put a significant challenge to players once its deterministic movement pattern is discovered. This paper addresses the problem of defining a smart, more challenging, and natural movement model for NPCs. Getting inspiration from the kids’ game “follow the leader” we adopt Artificial Intelligence techniques such as behavior trees and blackboards to provide NPCs with changing paths, dynamic aggregation in parties, and tactical decisions. Following this approach, players’ experience will greatly improve thanks to an always-changing battlefield scenario.

## 1 Introduction

In these recent years, with the gain in popularity of online games, and massive online games in particular, we are witnessing a progressive increment in the number of Massively Multiplayer Online Games (MMOGs) available over the Internet, as reported by (MMORPG community). A MMOG is a multiplayer game where a huge number of users in a shared virtual environment can interact in real time and band together to achieve shared goal such as conquering a map or defeating a group of monsters. When designing the environment, Non Playing Characters (NPCs) are the most common interaction points between players and the game system. NPCs are autonomous entities in the virtual world in charge of assigning tasks (quests) to players or to oppose players when they are trying to fulfill quests. In particular, mobs (shorthand for “mobiles”) are NPCs roaming the map and typically implement monsters to be defeated or avoided.

In standard implementations, due to technical and computational constraints, mobs (alone or in groups) patrol between pre-defined rally points on a map. This way, it is easy to keep mobs density uniform on the map and level designers may be reasonably sure about not having unguarded passages through a battlefield. Moreover, this standard approach is not very computational intensive and allows for a better scalability. Nevertheless, mobs moving between fixed rally points are not very entertaining for players (Koster, 2004): fixed paths can be easily guessed by experienced players and free passages in the battlefield will be discovered eventually. To address this issue, we believe that Artificial Intelligence

(AI) techniques can be exploited in order to provide a better user experience. By using AI, it is possible to model every mob as an independent agent; each agent may be able to randomly choose its own path while coordinating with others to keep mobs density constant.

This paper focuses on proposing and implementing a novel movement model for mobs inside an MMOG. NPCs will benefit from changing paths, dynamic aggregation in parties, and tactical decisions. The proposed solution will make gameplay more challenging and entertaining for players by providing always-changing battlefield scenarios.

The rest of this paper is organized as follows: first we discuss issues related to implementation of smart NPCs in Sec. 2; then, in Sec. 3, related work in literature is presented. The proposed solution is introduced and discussed in Sec. 4 while Sec. 5 provides details about our prototype implementation. Finally, Sec. 6 concludes the paper.

## 2 Artificial Intelligence and MMOGs

Artificial Intelligence has been present in videogames starting from the late 70s. Among the first examples we can find Qwak (Atari, 1974) and Pursuit (Atari, 1980). Pac-Man (Midwest Games, 1980) is the first notable example where the opponents, controlled by AI, follow distinct - and personalized - behaviors. Starting from Pac-Man, it become commonplace to include an AI subsystem in every game engine.

Videogame implementations up to a few years ago saw every agent as an independent entity taking autonomous decisions. This design was mainly due to the limited computational power available, constraining agents to very simple behaviors. In the majority of cases, NPCs were designed to just go toward or away from the player. Nevertheless, thanks to a new technology wave in the past decade, complex decision making techniques have been introduced in videogames. These decision making techniques allow for tactical coordination between agents with a significant increase in realism. Unfortunately, this same evolution is yet to come in MMOGs where the extremely high number of players and NPCs poses serious scalability limitations; World of Warcraft (Blizzard, 2004), as an example, reports 11.1 millions active users as of today.

When implementing an online game, one of the most important factors for player experience is the frame rate (Frame Per Second, FPS), which is the rate used to update the



game status on client side. A good, and constant, FPS value (typically between 30 and 60) provides a reactive game with smooth visuals. A reduction of FPS due to workload will lead to variable interaction times and unpredictable lags in the world evolution. In an MMOG, the update of the virtual world status is performed in a centralized way: a game server will take care to compute the new world configuration and distribute it through the network to all clients. To reach 30 FPS on every client, we are required to solve all interactions between users (and NPCs) at least once every 33 ms at server side. As a result, the number of NPCs and the complexity of their behavior is technically limited by the server computational power. Since the number of agents to simulate in an MMOG is at least two orders of magnitude greater than a single player game, AI solutions commonly used in standalone games becomes technically useless; game designers are then required to choose between few smart NPCs (in a small world) or many dumb ones (on a larger map).

Due to the above limitations, mobs' behavior in a MMOG follows a very simple model: every agents can assume two states: idle and in combat. When in idle state, the mob patrols between pre-defined rally points. If the mob is in combat state, it moves toward the player in a straight line and, when it gets close enough, starts attacking. The transition between these two states is triggered by the player's avatar being in or out the field of view of the mob. Usually, there is very little (if any) coordination between NPCs converging on the same player.

### 3 Related Work

Despite the fact that the problem we are addressing in this paper is of obvious interest for MMOG service providers, it seems, to the best of our knowledge, that limited efforts have been devoted to this topic from the scientific community. We may hint to two reasons for this situation: the tremendous expansion of online gaming is a relatively recent phenomenon, and commercial MMOG engines are usually secretly kept by the running companies making it very difficult for researchers to perform experiments. We believe that in the future, with the increasing popularity of open-engine MMOGs such as PlaneShift (Atomic Blue, 2000), the situation will positively change.

Most probably, the first work on the topic is (Reynolds, 1987). In this paper, the author proposes a computer model for a coordinated animal motion such as bird flocks and fish schools. These generic simulated flocking creatures have been baptized *boids*. While boids are actually providing a realistic behavior, they are missing any sort of group management: all agents in the simulation always move as a single flock.

In some more recent studies, such as (Synnaeve, 2010) and (Rhujittawiwat, 2006), AI techniques are applied to NPCs mainly to create a more believable behavior during combat. In (Synnaeve, 2010) authors use Bayesian programming to select actions and pick targets in a battlefield where both allies and foes are present while in (Rhujittawiwat, 2006) a genetic algorithm is used as a learning method to train an NPC about how to assist real players. None of these contributions are set on the specific goal to bound the computational workload on

servers and improve scalability. Moreover, both papers use simulated environments for performance analysis instead of real game engines.

Other works, such as (Combs, 2005) and (Fairclough, 2003), focus on using AI to improve storytelling. In particular, in (Fairclough, 2003) a character director system, which dynamically generates and controls a story, is proposed. In both cases, authors focus on storyline quality and completeness; limited attention is devoted to real-time interaction and system performances.

Most probably, the most interesting contribution on the topic comes from Zyda et al (Zyda, 2010a; Zyda 2010b). In these works, authors propose Cosmopolis: a free MMOG for larger-scale social modeling. In this virtual world, AI-driven NPC communities featuring customized cultural models are used as a means of researching interactions between individuals and societies. Despite some similarity in the goals, it is difficult to compare Cosmopolis and our work. This is because Cosmopolis is strictly focused on interaction and designed specifically as a research testbed for social and behavioral models while, in this paper, we propose an approach to be implemented in existing game engines to reduce computational workload on servers.

## 4 The Proposed Solution

In this section we are going to describe our solution to make mobs' movement and behavior more realistic in MMOGs.

To design our model we took inspiration from the kids' game "follow the leader". In this game a kid is elected as the leader and is free to move and behave as she likes; other participants are supposed to follow and mimic her, anyone who doesn't follow exactly is out of the game. Kids, during play, may also decide to join as followers or leave the group as they like.

We envision a MMOG where mobs move using a random path selection and are able to band together and perform coordinated actions. Every mob will wander randomly until another mob – or a group of mobs – will be in its field of view; when this happens, the wandering mob can decide to join to the mob/group and become a follower. In those cases where two mobs will band together, one of them will be elected leader and made responsible for group steering. Periodically, every member of the group may decide to leave and go back wandering alone; when the leader leaves the group another one will be elected.

From a technical point of view, the group may be considered as a stand-alone agent whose code sits on the leader; the group will be identified by means of a reference point (group position) and an orientation (see Fig. 1).

Following this approach, we can use a smart – and complex – behavior to manage the whole group while keeping the followers from using excessive computational resources. Resources request from the followers may get even lower than the standard approach thanks to the reduced number of decisions they will be required to take. Player experience, on the other hand, will benefit from many improvements:

- leaders (groups) paths may now be selected randomly to raise the bar in term of offered challenge;

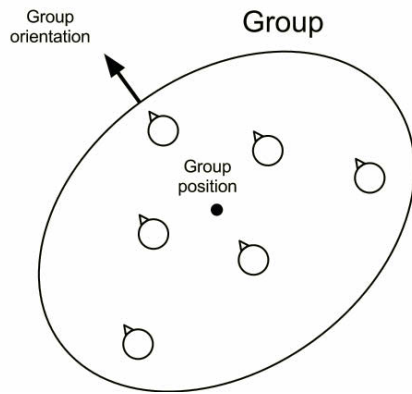


Figure 1: A group seen as a single agent with a leader and followers inside.

- players will feel the mobs as more realistic by seeing them moving in platoons;
- a group will be able to take tactical decisions based on a strategic assessment of player capabilities (i.e., to flee or to fight).

#### 4.1 Movement

The group movement, i.e., picking direction and speed for the group, will be in charge to the leader.

Many strategies are possible for path selection; e.g., we can choose a random rally point, pick a steering angle every frame, or head toward (or away from) other mobs or a point on the map. The implementation of these movement algorithms strictly depends on the kind of mobs we are going to model; as said, doing this at group level allows us to push on complexity without overloading the system. As long as the paths are not pre-determined our model will work and we are not going to discuss algorithms for path selection here.

The first real issue we are going to address is how to make all mobs move in a coordinated formation. As already explained, the leader will set a pace and all followers will do the same while trying to keep a certain distribution pattern. How the pattern is selected is also up to the kind of mobs we are going to model; as an example, if mobs are soldiers an ordered formation is preferred while if they are wild animals picking random positions around a center point might be a good solution.

A more important decision than selecting a distribution pattern is about choosing how group movement should be implemented. We have two options: one- and two-level formation steering. With one-level steering the pattern is fixed and the whole group moves as a single unit keeping rigid distances between followers. This solution is easier to implement but fitting the pattern in the environment (e.g., a pack of wolves traversing a forest) may become a problem. With two levels we have to provide a formation manager to process the pattern and assign destinations to followers (i.e., wolves will swerve around trees and go slightly out of formation but will also try to stay close to the ordered position). This second option is preferable because mobs will blend more nicely within the environment. Moreover, the additional computational requirement for individual

movements is, in total, usually less than what is required to fit the formation pattern in the environment.

To implement the two-level group movement we define a center point for the group (which might not be the physical location of the leader). Group collisions and line of sight may be calculated starting from this center point; this way, computation will not be required for every group member and having a global line of sight/hitbox will still be valid approximation. Mobs take relative positions around the center point following the required pattern and, on every frame calculation, a proper destination is assigned to each follower. All computation required from follower agents is a linear movement to location that is “close enough” to the current destination while avoiding environment obstacles.

#### 4.2 Group Management

The group management deals with members joining and leaving the group and how single actions are coordinated between group members. The most viable way to perform these operations is by means of a decision making process.

Implementing a decision making process in every group member is not an easy task. A first option could be to use a Finite-State Machine (FSM). A FSM is a mathematical model of computation used to design systems that can be in one of a finite number of states. The machine is in only one state at a time and can change from one state to another by a triggering event or condition. A FSM-based approach could be too demanding in term of memory and will require state changes depending on the state of other members, with severe scalability reduction. A second option could be to adopt Hierarchical FSM (HFSM), where the states can be ordinary states or super-states, which are FSM themselves. HFSM may be a better solution but they are not always granted to provide satisfactory performance. For the above reasons, we decided to adopt Behavior Trees (BTs) for group management. BTs are a formal, graphical modelling language used primarily in systems and software engineering; they employ a well-defined notation to unambiguously represent a large number of natural

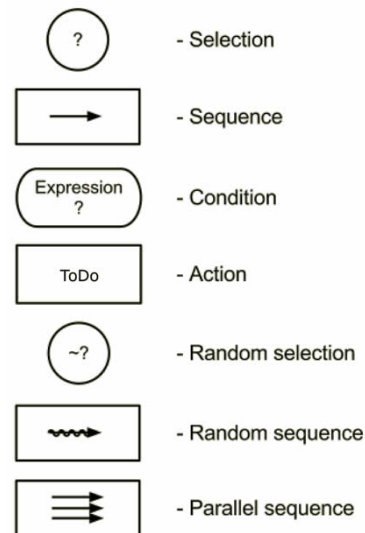


Figure 2: Symbols used in Behavior Trees.

language requirements to express the stakeholder needs for a software-integrated system. Thanks to their structure, BTs also allow to hide a subset of the information shared between group members in action and condition nodes; this way, we can also limit the complexity added by followers dynamically joining and leaving the group. Figure 2 reports the symbols we will be using in our BT schemes.

Communication and coordination between group members is performed using a blackboard system. A blackboard system is not a decision making tool in its own rather than a mechanism for coordinating the actions of several decision makers. The basic structure of a blackboard system has three parts: a set of different decision makers (called experts), a shared memory area, and an arbiter. Any expert may use the shared memory area to read system status and write suggestions about actions to be undertaken. Permission to access the shared memory must be granted by the arbiter. In our specific case, the experts are the mobs and the group agent.

Since, as already discussed before, we can see the whole group as a single agent we need two coordination systems: one between mobs and one between each mob and the group agent. To achieve this we extended the general concept of blackboard to a multi-level blackboard. A multi-level blackboard is a blackboard with two memory areas, which will host both of the above coordination systems (see Fig. 3). The first memory area is used for coordination between mobs; this area allows coordinating and prioritizing actions of single group members, e.g., first attack using magic, then with long-range weapons, and then using melee combat. The group agent uses the second memory area to suggest decisions to all mobs based on group status or environment; e.g., if one of the members is under attack someone must cure it and others should fight back.

An open issue with this blackboard-based approach is how to implement an arbiter that is able to satisfy the timing requirements of an MMOG. To speed-up execution, it is possible, when the number of experts is low, to remove the arbiter and use BTs where one or more nodes are tagged using states and priority values. During execution, first the group agent and then the other mob agents run their decision process. Once the group agent has run to completion it will write on the blackboard the selected action with its associated state and priority. After the group agent is done, all mob

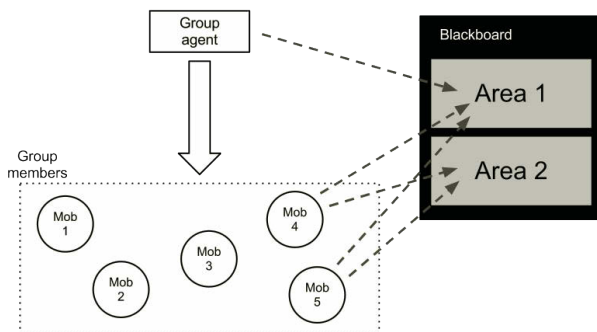


Figure 3: Scheme of a multi-level blackboard.

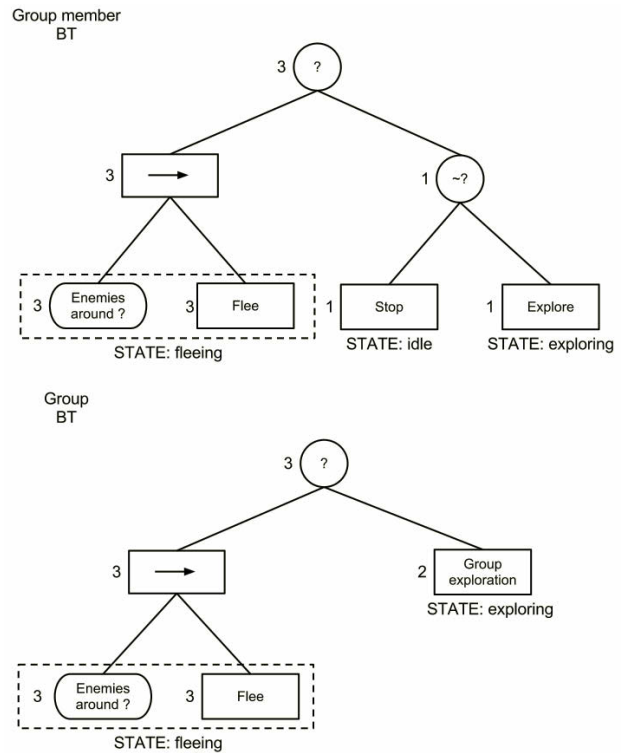


Figure 4: Sample Behavior Trees for an exploration group.

agents can start executing their own BTs. During execution, branches with a lower priority value than the one reported on the blackboard will not be evaluated. If, at the end of the execution a group member reaches a state with a higher priority than the one on the blackboard it will perform that action autonomously, otherwise (i.e., an untagged node has been reached or there is no execution with nodes on higher priority) it will perform the action selected by the group. In Fig. 4 a couple of sample BTs for an exploration group are shown. In the picture we can observe that explorers may act alone or in a group. In the case enemies are around the explorer will always flee. In the case no one is around it may decide to stay idle or to explore the surrounding, unless it is in a group; otherwise, it will be always required to explore with the other mobs (priority 2 on node “Group exploration”).

As we can see, following this approach it is easy to model complex behaviors for partially independent agents inside a group. Moreover, a single BT is able to describe both group and solo behavior for all mobs.

## 5 Prototype Implementation

In this section we are going to describe a prototype implementation of the system we presented in the previous section. This test application has been implemented using C++ and OpenGL.

In our prototype we define a square battlefield where a variable number of mobs and environment obstacles are randomly placed; see Fig. 5 for a screenshot. In the figure, the player's avatar is located in the lower right corner and the user can change its position using the keyboard. All other agents

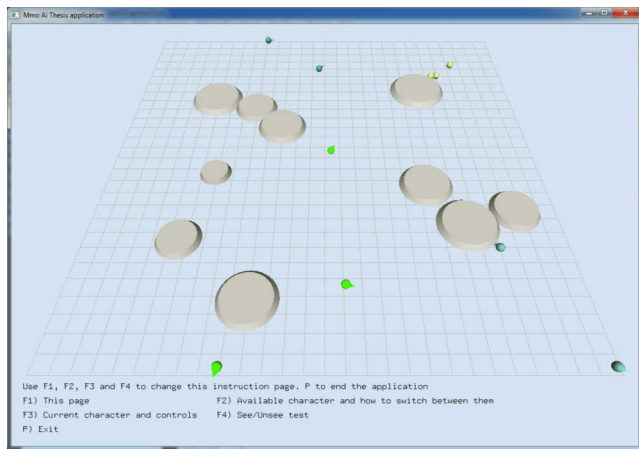


Figure 5: A screenshot of the prototype application.

are calculated in real time by our engine. To mimic an online game, the application implements two independent software modules (back-end and front-end) exchanging information via socket. The back-end works as a server taking care of AI execution for all agents and sending virtual world updates to the frontend. The front-end draws the virtual world on screen and implements the user interface. Using the front-end module the user can manipulate simulation parameters and move her avatar.

Mobs can be assigned to different groups (indicated by their color in the screenshot); only mobs belonging to the same group can band together. These groups are modeling different kind of monsters sharing the battlefield. Mobs will not attack each other.

When a group encounters the user's avatar a tactical decision is taken by the leader based on the group strength (the number of members): if the group size is below a certain threshold (i.e., it is weaker than the user) it will start to flee, otherwise it will start an attack. Since the combat system is not a key point at this stage of the project, attackers will just go toward the target and the defender is supposed to flee. During pursuit, the number of group members may change, because mobs can still join, leave, or are just left behind. On a group change, the tactical decision is re-taken and an attacking group may decide to just leave the player alone. If the player manages to leave behind a pursuing group, the mobs will start moving randomly again.

The purpose of environment obstacles is mainly to check the two-level implementation of group movement. They are currently implemented as circular “hills” to achieve easier collision detection based on distance.

## 5.1 Movement

Every group (and single mob) moves at a constant speed along its orientation. On every frame calculation a (small) steering angle is randomly selected taking collisions with hills into account. The group will move using an arrowhead formation (as in Fig. 1).

## 5.2 NPCs Behavior Tree

The BT used in mobs is depicted in Fig. 6. As we can see, the first branch is selected when the mob senses an enemy (or is under attack) or there is a help request from another mob nearby. If the mob is inside a group it will enter the “Notifying” state and write on the blackboard that an enemy is

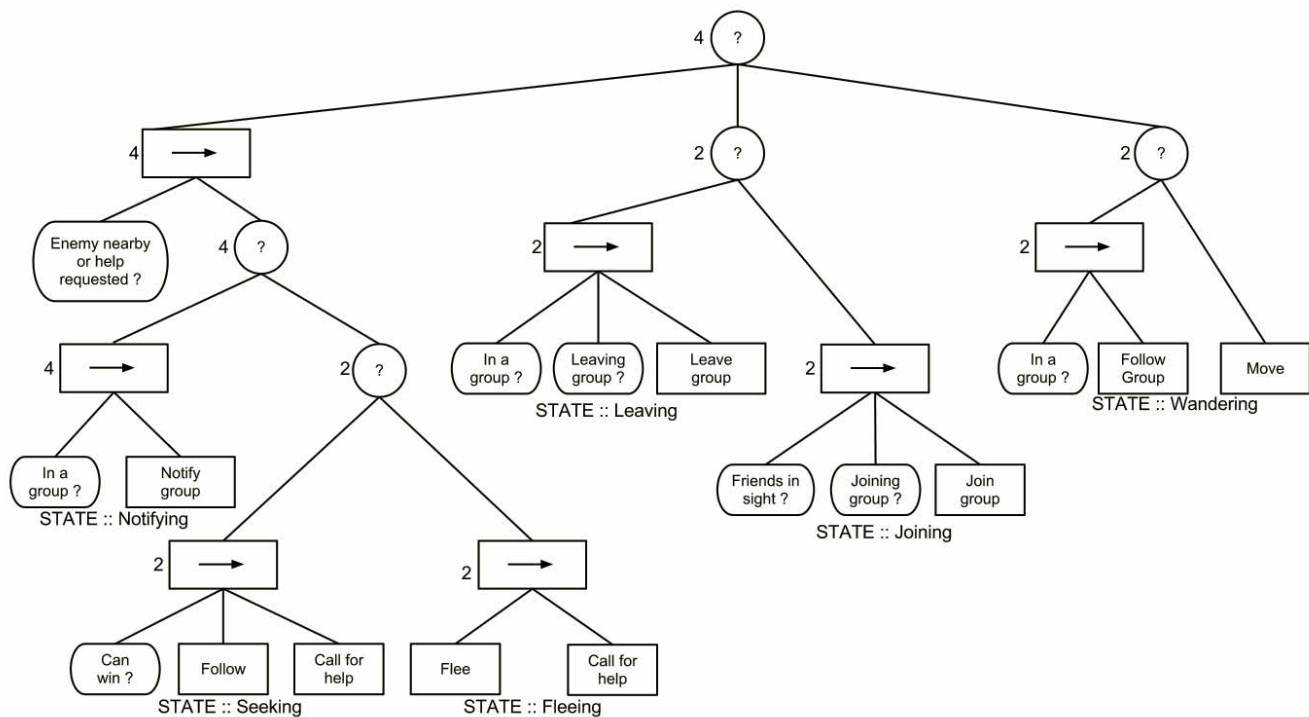


Figure 6: NPCs Behavior Tree in the prototype.



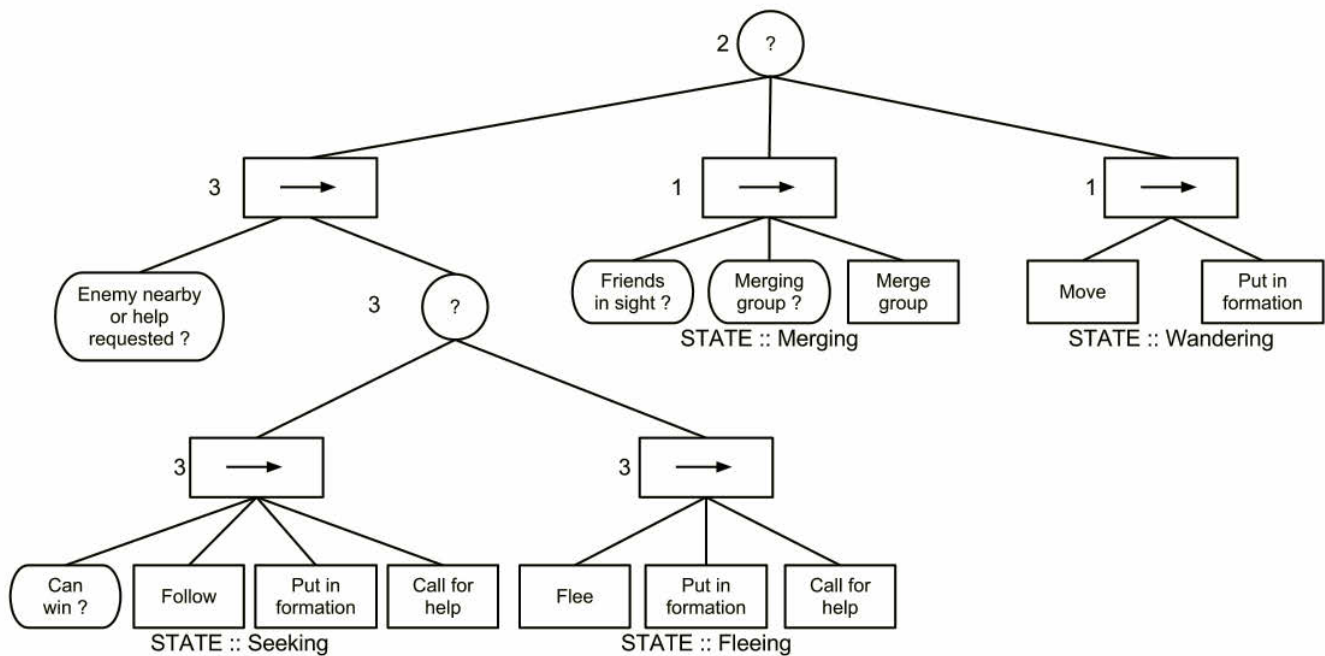


Figure 7: Group Behavior Tree in the prototype.

nearby and submit to the tactical decision that will be taken from the leader. Posting the message on the blackboard will also allow other group members to converge on the enemy even if it is not yet in their field of view. Otherwise, if the mob is not in a group, it must take its own tactical decision: fight or flee. If the mob evaluates it can win the battle it will start in pursuit (“Seeking” state) of the enemy, calling also for reinforcements. Otherwise, if the enemy is stronger, it will start to escape (“Fleeing” state) also calling for reinforcement and, eventually, to become stronger and fight back. The branch we just discussed will be evaluated also in the case the mob will hear a help request coming from another group (if it is from the same group, the blackboard takes precedence). When called for assistance, the mob will still need to take a tactical decision and evaluate if it is better to give support or flee.

The second branch of the behavior tree deals with group management. If the mob is in a group and wants to leave the state is set to “Leaving”; then, the mob will become an independent agent. Otherwise, if there are friendly units nearby and the mob wants to join the group, the state is set to “Joining”. The willingness of a mob to join or leave a group is evaluated with a random variable following a Bernoulli distribution where  $p$  can be set from the front-end interface (by default  $p=0.1$ ). In order to limit group dynamics we also set a minimum time  $t$  for the mob to stay in a group before leaving: only if enough time has passed the random variable can be evaluated. As for  $p$ ,  $t$  is set from the interface with a default value of 5 seconds.

The last branch is the one managing movement when no other mob is around (“Wandering” state). If the mob is part of a group it will just follow the leader (reading from the blackboard); otherwise, it will pick its own random steering angle and proceed along its orientation.

### 5.3 Group Behavior Tree

The BT used in groups is presented in Fig. 7. The first branch in the figure, as for the mobs, is selected when there are enemies around. Differently from the previous case, the group will not post on the blackboard but, rather, force a (global) strategic decision on all group members with priority 3. Single mobs will still notify other group members (priority 4) but will follow the strategic decision taken from the group agent, because that specific sub-branch has priority 2. When the tactical decision is taken from the group agent, the “Seeking” and “Fleeing” states also provide an action labeled “put in formation”. This action will make all group members rearrange around the group position; the actual location may change depending on the mob stance: attacking, fleeing, or just moving around.

The second branch manages group merging: if two groups are close to each other, one of the group agents may decide to merge and become one with the other. This state has no correspondence with the “Joining” and “Leaving” states of an NPC; since it has a lower priority compared to the second branch in Fig. 6, it is possible for a mob to leave a group independently from an ongoing merging action.

The last branch of the BT takes care of the group movement when no one is around: it sets the new group position (using the blackboard) and puts all participants into formation.

### 5.4 Testing the Architecture

During experiments the user moves on the battlefield and try to escape from the mobs after attracting their attention. Subjective evaluation of groups in pursuit gave positive results in term of movement patterns: formation seems to

adapt smoothly enough to the obstacles and server workload is reasonable. As a matter of fact, CPU usage never rose over 70%.

Increasing the number of mobs on the map requires more than 500 agents to drop performance below 30 FPS on a middle-end PC (Intel dual core at 2.5 GHz with 4 GB of main memory). Despite the fact that 500 agents are much less than the total population of an MMOG (which is in the order of millions, as claimed in Sec. 2), we have to remember that: (i) agents are managing NPC, which are outnumbered by players with at least two order of magnitude and (ii) the complete infrastructure of an MMOG provides a virtual world replicated on a number of independent clusters (realms) where each server in a cluster is in charge of a subsection of the local map. With the above considerations in mind, an acceptable workload with 500 agents is also an encouraging result. Nevertheless, the scenario is still hard to evaluate without an implementation inside an actual game engine.

## 6 Conclusion and Future Work

In this work we addressed the problem of realistic movement for Non Playing Character in Massively Multiplayer Online Games. Realistic behavior can be achieved adopting Artificial Intelligence techniques but, despite tremendous technical improvements in the last decade, application to massive environments is still critical due to performance constraints. We proposed a solution based on a hierarchical approach: NPC can aggregate in groups to be managed as a single agent; a group agent will impose general decisions (such as movement) to all members while single NPCs will manage simpler decisions (such as joining or leaving a group). The proposed solution has been used to implement a prototype. In our prototype NPCs populate a battlefield with environment obstacles; they roam the battlefield and eventually aggregate in groups to attack (or escape from) the user. First results are encouraging: NPCs behavior feels quite natural while server workload is bound to an acceptable level.

Future evolutions for this work include cooperation between heterogeneous NPCs, and the implementation inside a real game engine such as unity3d (Unity Technologies, 2005) or smartfox (gotoAndPlay(), 2004).

## References

- Atari (1974) Qwak!. [http://en.wikipedia.org/wiki/Qwak!\\_\(arcade\\_game\)](http://en.wikipedia.org/wiki/Qwak!_(arcade_game)). Last visited Apr. 13, 2013.
- Atari (1980) Pursuit. [http://en.wikipedia.org/wiki/Pursuit\\_\(video\\_game\)](http://en.wikipedia.org/wiki/Pursuit_(video_game)). Last visited Apr. 13, 2013.
- Atomic Blue (2000). Planeshift. <http://www.planeshift.it/>. Last visited Apr. 13, 2013.
- Blizzard (2004) World of Warcraft. <http://www.worldofwarcraft.com/>. Last visited Apr. 13, 2013.
- Combs, N. (2005). The Intelligence in the MMOG: from scripts to stories to directorial AI. In Proc. of Other Players Conference, IT University, Copenhagen.
- Fairclough, C. and Cunningham, P. (2003). A Multiplayer Case Based Story Engine. Dublin, Trinity College Dublin, Department of Computer Science, TCD-CS-2003-43.
- gotoAndPlay() (2004). Smartfox server. <http://www.smartfoxserver.com/>. Last visited Apr. 13, 2013.
- Koster, R. (2004). A Theory of Fun for Game Design. Paraglyph Press.
- Midwest Games (1980) Pac-Man. [http://en.wikipedia.org/wiki/Pac\\_man](http://en.wikipedia.org/wiki/Pac_man). Last visited Apr. 13, 2013.
- MMORPG Community. <http://www.mmorpg.com/>. Last visited Apr. 13, 2013.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In Proc. 14<sup>th</sup> annual conference on Computer graphics and interactive techniques, Anaheim, California.
- Rhujittawit, T. and Kotrajaras, V. (2006). Learnable Buddy: Learnable Supportive AI in Commercial MMORPG. In Proc. 9<sup>th</sup> International Conference on Computer Games: AI, Animation, Mobile, Educational & Serious Games, Dublin, Ireland.
- Synnaeve, G. and Bessière, P. (2010). Bayesian Modeling of an Human MMORPG Player. In Proc. 30<sup>th</sup> International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Chamonix, France.
- Unity Technologies (2005). Unity3D game engine. <http://unity3d.com/>. Last visited Apr. 13, 2013.
- Zyda, M., Spraragen, M., Ranganathan, B., Arnason, B. and Landwehr P. (2010a). Designing a Massively Multiplayer Online Game/Research Testbed Featuring AI-Driven NPC Communities. In Proc. of 6<sup>th</sup> AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, California.
- Zyda, M., Spraragen, M., Ranganathan, B., Arnason, B. and Liu, H. (2010b). Information Channels in MMOGs: Implementation and Effects. In Proc. 1st International Conference on Human Factors and Ergonomics in Healthcare, Florida.

# Analyzing Program Evolution in Genetic Programming using Asynchronous Evaluation

Tomohiro Harada<sup>1,2</sup> and Keiki Takadama<sup>1</sup>

<sup>1</sup>The University of Electro-Communications, Japan

<sup>2</sup>Research Fellow of the Japan Society for the Promotion of Science DC  
harada@cas.hc.uec.ac.jp

## Abstract

This paper investigates the evolution ability of Tierra-based Asynchronous Genetic Programming (TAGP) as GP using an *asynchronous* evaluation. We compare TAGP with two simple GP methods, steady-state GP and GP using  $(\mu + \lambda)$ -selection as GP using a *synchronous* evaluation. Three GP methods are compared in experiment to minimize the size of an actual assembly language program in several computational problems, two arithmetic and two boolean problems. The intensive comparisons have revealed the following implications: (1) TAGP has higher evolution ability than GP using synchronous evaluation, *i.e.*, TAGP can evolve smaller size programs which cannot be evolved by GPs using synchronous evaluation; and (2) the diversity of the programs evolved by TAGP can derive a high evolution ability in comparison with GP using synchronous evaluation.

## Introduction

*Evolutionary Algorithms (EAs)* like *Genetic Algorithm (GA)* (Goldberg, 1989) and *Genetic Programming (GP)* (Koza, 1992) requires the appropriate diversity of a population to evolve solutions efficiently. As approaches to such diversity, the conventional EAs focused on the genetic operators or selection strategies. For example, the adaptive parameter setting methods have been proposed to control a probability of the genetic operators depending on the fitness in the population (Subbu et al., 1998; Yun and Gen, 2003; Lin and Gen, 2009), or NSGA-II (Deb et al., 2002) which is well known multi-objective evolutionary algorithm (MOEA) employs an idea of the crowding distance which is based on the distance of solutions to maintain the diversity of solutions. Furthermore, EAs such as *Differential Evolution (DE)* (Storn and Price, 1997) and *MOEA/D* (Zhang and Li, 2007) which have recently attracted much attention on have a high evolution ability by evolving solutions independently, which contributes to maintaining appropriate diversity of solutions. Such an independent evolution is based on the *asynchronously* evaluation approach which evolves solutions asynchronously unlike the conventional EA approach which is based on the *synchronous* evaluation approach which evolves solutions

*synchronously*, *i.e.*, solutions are evolved by genetic operators after all individuals are evaluated. One main advantage of such *asynchronous evaluation* in EAs is to be able to derive the diversity of a population without any special heuristic operation.

This paper aims at verifying the evolution ability of the asynchronous evaluation on the program evolution. Since the previous asynchronous approaches such as DE and MOEA/D cannot be easily applied to the program evolution, this paper employs a novel GP method using the asynchronous evaluation, named as *Tierra-based Asynchronous Genetic Programming (TAGP)* (Nonami and Takadama, 2007; Harada et al., 2010, 2011) the previous researches proposed. TAGP is based on the idea of a biological evolution simulator, Tierra (Ray, 1991) and asynchronously evaluates and evolves programs. Since TAGP has the same advantage of the other EAs using asynchronous evaluation such as DE and MOEA/D, TAGP has a potential of evolving programs efficiently by maintaining the appropriate diversity of a population. To investigate such as evolution ability of TAGP, this paper compares TAGP as GP using the asynchronous evaluation, with two simple GP methods, steady-state GP (SSGP) (Reynolds, 1993) and GP using  $(\mu + \lambda)$ -selection  $((\mu + \lambda)$ -GP) as GP using the synchronous evaluation. The experiment applies these three GP methods to several computational problems to minimize the size of an actual assembly language program.

In the following section, we firstly explain a biological evolution simulator, Tierra, which ideas are employed in TAGP, and explains the algorithm of TAGP. Then we compares TAGP with SSGP and  $(\mu + \lambda)$ -GP in several computational problems, and gives its result and detailed analyses. Next section discusses the difference of three GP methods, and this paper finally gives conclusions and future works.

## Tierra

Tierra (Ray, 1991) proposed by T. S. Ray is a biological evolution simulator, where digital creatures are evolved through a cycle of a self-reproduction, deletion and genetic operators such as a crossover or a mutation. Digital creatures live

in a memory space corresponding to the nature land on the earth, and they are implemented by a linear structured computer program such as the assembly language to reproduce (copy) themselves to a vacant memory space. CPU time corresponding to energy like actual creatures is given to each creature, and they execute instructions of a self-reproduction program within allocated CPU time. Since given CPU time is shorter for execution time of programs, all programs are executed in parallel. Lifespan of a program is decided with a *reaper* mechanism. All programs are arranged in a queue, named as *reaper queue*, and a reproduced program is added to the end of the reaper queue. While program execution, a program that can correctly execute its instruction moves its position in the reaper queue to lower, while one that cannot correctly execute its instruction moves its position to upper. Then, when a memory space is filled, a program that is at a top of the reaper queue is deleted from the memory. Due to the reaper mechanism, programs that cannot reproduce themselves within allocated CPU time or include some incorrect instructions are deleted from the memory, while creatures that can reproduce themselves propagate in the memory.

As results of such evolution, Tierra generates, for example, programs, called *parasite*, that reproduce themselves by using other program's instructions, or ones, called *hyper-parasite*, that have immunity to the *parasites*. Note that this evolution is not pre-programmed in Tierra but is caused by *emergence* (Langton, 1989). As the final stage of Tierra, programs that have shorter program size or have efficient algorithm are generated, that require less CPU time than an initial program to reproduce themselves (ATR Evolutionary Systems Department, 1998).

## Tierra-based Asynchronous Genetic Programming

### Overview

The previous researches focus on the feature of Tierra which can evolve programs, *i.e.*, digital creatures, with asynchronous execution, and have proposed a novel GP based on Tierra mechanism, named as *Tierra-based Asynchronous Genetic Programming (TAGP)* (Nonami and Takadama, 2007; Harada et al., 2010, 2011). To apply Tierra to evolving programs with a given task, the previous research introduces *fitness* commonly used in EAs to evaluate programs, and also introduces reproduction and deletion mechanisms depending on *fitness* into Tierra. This is because it is impossible to give any purposes to programs in Tierra whose purpose is only to reproduce themselves.

Fig. 1 shows an image of TAGP. TAGP firstly starts from a program that completely accomplishes the given task. Programs that consist of a linear structured instructions and some registers are stored in a limited memory space. Each program executes a small number of instructions, which is

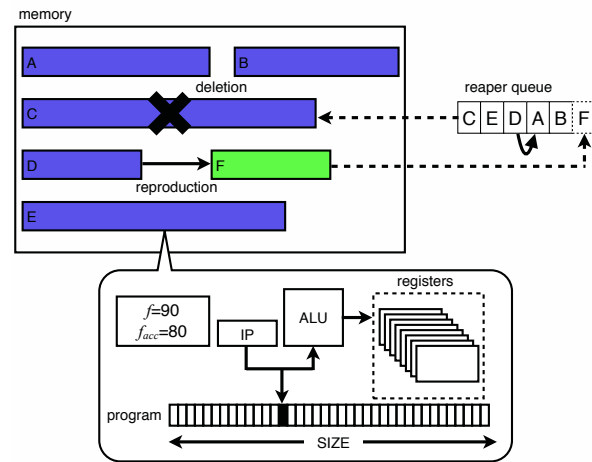


Figure 1: An image of TAGP

preconfigured, *e.g.*, three instructions, to simulate a parallel execution. All programs are arranged on *reaper queue* that controls lifespan of programs. When an execution of a program is finished, its fitness is evaluated depending on its execution result, and the reproduction and the reaper queue control are asynchronously conducted according to its fitness. When the memory is filled with programs, programs that are arranged at the upper of the reaper queue are removed from the memory.

### Algorithm

TAGP evolves programs through the following selection, reaper queue control, reproduction, and deletion algorithms. The algorithm of TAGP is shown in Algorithm 1 where 1,  $prog.f_{acc}$  and  $prog.f$  respectively indicate accumulated and evaluated fitness, and  $rand(0, 1)$  indicates random real value between 0 to 1.  $prog_{prev}$  indicates a previously selected program, while  $elite_{prev}$  indicates a previously selected *elite* program (detailed in below).

**Selection and reaper queue control** When an execution of one program is finished, its fitness is evaluated depending on its register value and the evaluated fitness is added to an *accumulated fitness*  $prog.f_{acc}$  (the 1<sup>st</sup> line in Algorithm 1). Based on  $prog.f_{acc}$ , whether a program is selected as a reproduction candidate or not is determined. Let represent the maximum fitness as  $f_{max}$ , if the accumulated fitness of a program exceeds  $f_{max}$ , it is selected as a reproduction candidate, and  $f_{max}$  is subtracted from its accumulated fitness (the 2<sup>nd</sup> and 3<sup>rd</sup> lines). While if not, a program is not selected. Depending on this selection condition, a program that completely accomplishes the given task, *i.e.*, its fitness is equal to  $f_{max}$ , is invariably selected because the accumulated fitness always exceeds  $f_{max}$ . High fitness programs have a high potential to be selected because the accumulated



**Algorithm 1** The algorithm of TAGP

---

```

1:  $prog.f_{acc} \leftarrow prog.f_{acc} + prog.f$ 
2: if  $prog.f_{acc} \geq f_{max}$  then
3:    $prog.f_{acc} \leftarrow prog.f_{acc} - f_{max}$ 
4:   repeat
5:     down reaper queue position
6:   until  $rand(0, 1) < P_{down}(prog.f)$ 
7:   reproduce better program of  $prog$  and  $prog_{prev}$  with
   genetic operators
8:    $prog_{prev} \leftarrow prog$ 
9:   if  $prog.f = f_{max}$  then
10:    if  $prog$  is better than  $elite_{prev}$  then
11:      reproduce  $prog$  without any genetic operators
12:    else
13:      reproduce  $prog$  with genetic operators
14:    end if
15:     $elite_{prev} \leftarrow prog$ 
16:  end if
17: else
18:   if  $rand(0, 1) < rand(\frac{prog.f}{f_{max}}, 1)$  then
19:     remove  $prog$  from memory
20:   end if
21:   repeat
22:     up reaper queue position
23:   until  $rand(0, 1) > P_{up}(prog.f)$ 
24: end if

```

---

fitness frequently exceeds  $f_{max}$ , while low fitness ones are hard to satisfy this condition.

Then, a position in the reaper queue of a program that satisfies the selection condition becomes lower than the current one, *i.e.*, its deletion probability decreases (the 4<sup>th</sup> ~ 6<sup>th</sup> lines), while one that does not satisfies the condition becomes upper, *i.e.*, its deletion probability increases (the 21<sup>st</sup> ~ 23<sup>rd</sup> lines). The move distance is determined by the move rate represented as  $P_{down}$  and  $P_{up}$  which are calculated as the following equation based on fitness,

$$P_{down}(f) = \frac{f}{f_{max}} \times P_r, \quad P_{up}(f) = \frac{f_{max} - f}{f_{max}} \times P_r, \quad (1)$$

where  $P_r$  is the maximum probability of  $P_{down}$  and  $P_{down}$ , which is preconfigured. Depending on these equations, higher fitness programs are arranged on lower position in the reaper queue, *i.e.*, survive long, while lower ones are arranged on upper, *i.e.*, are easily removed.

**Reproduction** Programs selected depending on the selection condition become a reproduction candidate. To reproduce better program asynchronously, TAGP only compares two programs, a currently selected program and a previously selected program described as  $prog_{prev}$  in Algorithm 1 and better one is selected as a parent. A selected program generates an offspring with the genetic operator such as a

crossover and a mutation, and an offspring is reproduced to a vacant memory space that is larger than its program size (the 7<sup>th</sup> ~ 8<sup>th</sup> lines). Additionally, the elite preserving strategy (Jong and Alan, 1975) is applied to preserve programs that can accomplish the given task (the 9<sup>th</sup> ~ 16<sup>th</sup> lines). If a current program is evaluated as  $f_{max}$ , it is compared with one that is previously evaluated as  $f_{max}$  represented as  $elite_{prev}$  in Algorithm 1. Then if the current one is better, it is reproduced as an elite program *without* the genetic operators to preserve better program, *i.e.*, generating a copy of the elite, while if not, it is reproduced *with* the genetic operators. TAGP employs four genetic operators, a crossover, a mutation, and an instruction insertion/deletion. The crossover operator combines a reproduced program with a previously selected parent. The mutation operator changes one random instruction in a reproduced program to other random instruction. The insertion operator inserts one random instruction into a reproduced program, while the deletion operator removes one instruction selected at random in a reproduced program.

**Deletion** TAGP conducts two deletion. One is a deletion based on the reaper queue that is conducted during the reproduction process. If a vacant memory space is not found when reproducing an offspring, programs that is arranged upper in the reaper queue are removed until a total vacant memory space becomes greater than a certain threshold, *e.g.*, usually set as 20% of the memory. This deletion remove elder and lower fitness program.

While another deletion is a *natural death* which is based on the idea of *sugarscape* (Epstein and Axtell, 1996). The natural death applied to programs that do not satisfied the selection condition according to the 18<sup>th</sup> ~ 20<sup>th</sup> line in Algorithm 1, where  $rand(a, 1)$  indicates random real value between  $a(\leq 1)$  to 1. This deletion removes lower fitness programs even if the memory is not filled.

## Experiment

To validate the evolution ability of TAGP, this paper compares TAGP with two simple GP methods, steady-state GP (SSGP) (Reynolds, 1993) and GP using  $(\mu + \lambda)$ -selection  $((\mu + \lambda)$ -GP), with four computational problems. SSGP and  $(\mu + \lambda)$ -GP are hereinafter collectively called SGPs (synchronous GPs). SSGP selects two parents from the population and generates two offspring, then the worst two programs in the population are replaced with generated offspring.  $(\mu + \lambda)$ -GP generates  $\lambda$  offspring from the population consisting of  $\mu$  programs, and leaves better  $\mu$  programs from  $(\mu + \lambda)$  programs to next generation, where  $\mu = \lambda$  in this experiment.

## Computational problems

This paper applies three GP methods to the following four computational problems shown in Table 1.

Table 1: Computational problems

Problem type	Function	# of training data
Arithmetic1	$x^4 + x^3 + x^2 + x$	16
Arithmetic2	$x^y$	25
Boolean2	6-multiplexer	64
Boolean2	5bits digital adder	32

Fitness is evaluated as

$$fitness = f_{max} - \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|, \quad (2)$$

where  $f_{max}$  indicates the maximum fitness,  $n$  indicates the number of the training data,  $\hat{y}_i$  indicates the function value calculated from the  $i^{th}$  training data, while  $y_i$  indicates the output of a program in respect to the  $i^{th}$  input value. Note that when comparing the same fitness programs, the program size is firstly compared, and if is also equal to, the number of executed instructions to calculate all training data is finally compared.

This paper employs a program written by an actual assembly language embedded on PIC16 micro-controller unit (Microchip Technology Inc., 2007) developed by Microchip Technology Inc.. This is 12bits word assembly language, and has 33 simple instructions that consist of add-subtract, logical, bit, and branch instructions, but not contain a multiplication. One program can use 16 general 32-bits registers, named as  $R0$  to  $R15$ , and one temporary 32-bits register, named as  $W$ , while its size is limited to 256 instructions. The input value is firstly set from  $R1$  register, and other registers are initialized as 0. Note that since this instruction set does not include a multiplication instruction, programs have to combine some instructions and loop structures to calculate multiplication. Therefore, to calculate both of regression problems, programs have to include loop structures. Concretely, Arithmetic1 includes three multiplications ( $x^2$ ,  $x^3$ , and  $x^4$ ), while Arithmetic2 includes double multiplications loop to calculate  $x^y$  in both initial programs.

The experiment starts from an initial program that can completely solve the given task, and compares evolution ability of three GP methods by observing how small program can be obtained in one hour.

### Parameter settings

Common parameter settings for all three GP methods is shown in Table 2, ones for only SGPs is shown in Table 3, and ones for only TAGP is shown in Table 4. Three GP methods employ the same genetic operators, crossover, mutation, and instruction insertion and deletion, and also employ the same parameters for the crossover, mutation, insertion, and deletion rate. Only one genetic operator is executed with the configured probability for each reproduction. Crossover method is two point crossover, and the maximum

Table 2: Common parameter settings

Parameter	value
Crossover rate	0.7
Mutation rate	0.1
Insertion rate	0.1
Deletion rate	0.1
Crossover method	Two point crossover
$f_{max}$	100

Table 3: Parameter settings of SSGP and  $(\mu + \lambda)$ -GP

Parameter	value
Selection	Binary tournament
Upper execution steps	50000

Table 4: Parameter settings of TAGP

Parameter	value
Removing threshold	20% of memory
$P_r$	0.9

Table 5: Memory size of TAGP and population size of SSGP and  $(\mu + \lambda)$ -GP in each problem

Problem type	TAGP	SSGP	$(\mu + \lambda)$ -GP
Arithmetic1	6400	100	100
Arithmetic2	6400	200	50
Boolean1	25600	200	200
Boolean2	6400	50	200

fitness ( $f_{max}$ ) is set as 100. In SGPs, the binary tournament selection is employed, while the upper execution steps are restricted to 50,000 and if execution steps exceeds, its fitness becomes the minimum value. In TAGP, the deletion removes programs until the total vacant memory exceeds 20% of the memory size, while  $P_r$  is set as 0.9, which is the maximum probability of  $P_{down}$  and  $P_{up}$ .

Settings of population size of SGPs and memory size of TAGP is determined based on pre-experiment. SGPs compare results of 50, 100, and 200, while TAGP compares results of 6400, 12800, and 25600. Note that since the maximum program size is configured as 256 instructions, when the population size is set as 100, a used memory space of SGPs is equal to the memory size 25600(= 256 (instructions)  $\times$  100 (individuals)) of TAGP.

### Result

The experiment conducts 30 trials for three GP methods, and we evaluate three GP methods based on a percentage of trials in which a program for each size can be generated in one hour. Fig. 2 shows the result of a percentage of trials in which a program for each size can be generated in one hour. In Fig. 2, the abscissa indicates the program size, while the ordinate indicates the percentage of trials which can generate a program for each size. The red lines show the result

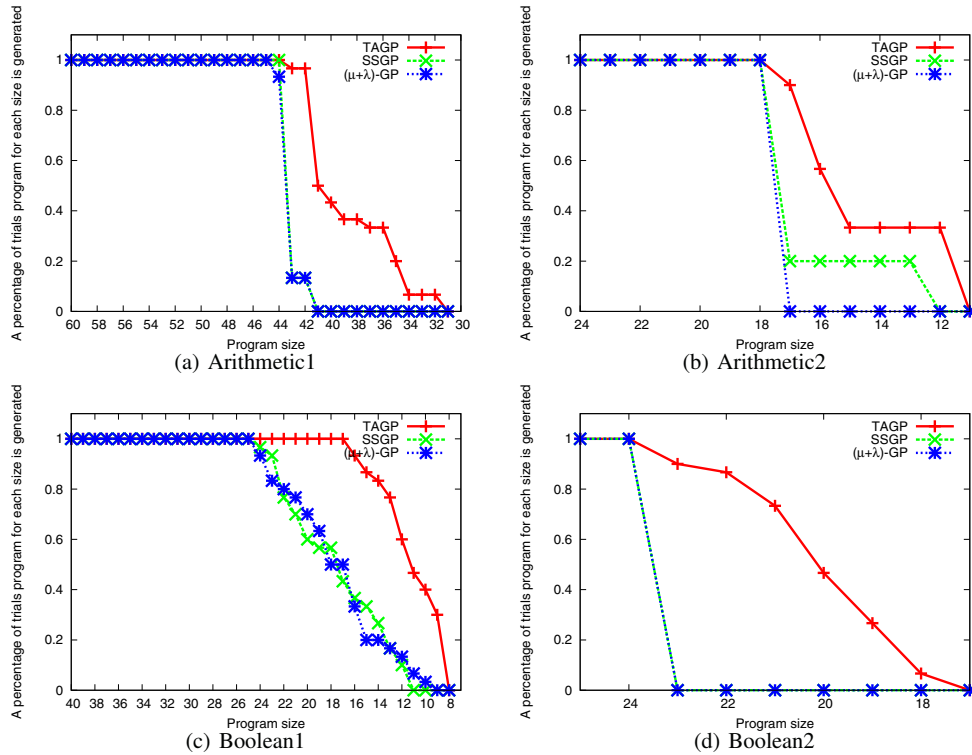


Figure 2: A percentage of trials in which a program for each size is generated in one hour

of TAGP, the green lines show the result of SSGP, while the blue lines show the result of  $(\mu + \lambda)$ -GP.

From Fig. 2, it is revealed that, in all computational problems, TAGP can generate smaller size programs than both of SGPs. Mann-Whitney U test is used to compare the generated program size of all GPs in all problems. The level of significance is set at  $\alpha = 0.05$ , and significant differences between TAGP and SGPs are verified. Focusing on the program size finally generated with each GP, programs of a certain size are generated with all GPs, concretely 44, 18, and 24 in Arithmetic1, Arithmetic2, and both of boolean problems respectively. It is, however, hard for SSGP and  $(\mu + \lambda)$ -GP to generate programs that size is smaller than the size described above. While TAGP can generate such programs in most trials.

To clarify the reason why TAGP outperforms SGPs, in the following, we analyze evolution processes observed in experiments. The evolution processes falls into two main categories, *non-destructive evolution* and *destructive evolution*. *Non-destructive evolution* means that it does not affect calculation result, *i.e.*, does not decrease fitness of a program, during evolution process, and is easily achieved. While *destructive evolution* means that it is required to decrease fitness of a program or to increase program size during evolution process. Non-destructive evolution also falls

into two categories, *single-step* and *multi-step*, the former means that it completes only single genetic operation, while the later means that it requires two or more genetic operations to complete its evolution process. Note that all destructive evolution is *multi-step* because it requires two or more genetic operations. The following sections describe details of these evolutions.

**Single-step non-destructive evolution** Single-step non-destructive evolution evolves programs by removing unnecessary instructions which do not affect the calculation result. As shown in Table 6, the problems except Boolean2 include some number of unnecessary instructions, and removing these instructions can decrease the program size. This evolution can be easily achieved because it can decrease the program size by not affecting the calculation result, *i.e.*, not decreasing fitness. Regarding Arithmetic1, Arithmetic2, and Boolean1, since programs that size is respectively 44, 18, and 36 are easily generated by only removing unnecessary instructions, all GPs accomplish 100% success rate, but one trial of  $(\mu + \lambda)$ -GP in Arithmetic1.

**Multi-step non-destructive evolution** Multi-step non-destructive evolution generates small size programs by replacing several instructions with one or a few instructions. Examples of observed evolutions are shown in Fig. 3. In

Table 6: The initial program size and the number of unnecessary instructions in each problem

Problem type	initial program size	# of unnecessary instructions
Arithmetic1	62	18
Arithmetic2	25	7
Boolean1	41	5
Boolean2	26	0

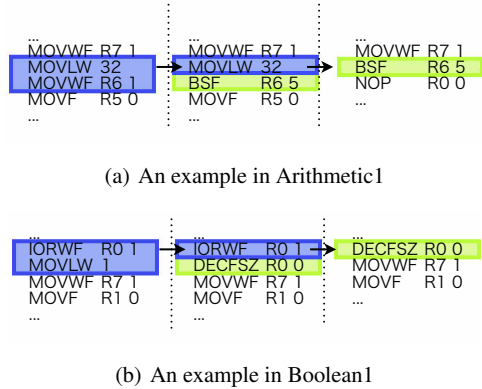


Figure 3: An example of multi-step non-destructive evolution

these figures, colored instructions are replaced with one instruction by using two genetic operations.

In Fig. 3, the left figure indicates a part of a program before evolutions, while the center and the right ones indicate the change of instructions. The colored instructions in the left figure substitutes a loop counter 32 to  $R6$  register through the temporary register  $W$ . This process requires two instructions, *MOVLW* which substitutes a literal to  $W$  register and *MOVWF* which substitutes the value of  $W$  register to any register. These two instructions are replaced with *BSF* which sets 1 to any one bit of any register. In this case *BSF* sets 1 to 5<sup>th</sup> bit of  $R6$  register. This works just as well as the previous two instructions, *i.e.*, sets 32 to  $R6$  register. While in Boolean1, a lot of examples are observed to replace logical instructions (*AND*, *OR*, and *XOR*) with conditional branch instructions. In example of Fig. 3(b), *OR* instruction (represented as *IORWF*) is replaced with *DECFSZ* instruction. *DECFSZ* instruction decrements a register value and skip next instruction only if its result is equal to 0. Such replacement of logical instructions with branch instructions are often observed both of boolean problems, this is because most of boolean calculation can be calculated by using conditional branches.

Since multi-step non-destructive evolutions does not also affect the calculation result in their evolution process, it is possible to decrease the program size by sequential genetic

operations. In particular, programs that size is smaller than 36 in Boolean1 and ones that size is 24 in Boolean2 are generated through this evolution.

**Multi-step destructive evolution** Although multi-step destructive evolution also generates small size programs by replacing several instructions with one or a few instructions such as multi-step non-destructive evolution, the selection probability of the program decreases in the process of evolution, *i.e.*, fitness decreases or program size increases. It is difficult to achieve this evolution because programs that are in the process of evolution can be removed from the population before generating small size programs. Examples are shown in Fig. 4, where colored instructions have same meaning in previous figures.

In an example of Arithmetic1, shown in Fig 4(a), a loop structure that calculates  $x^4$  with  $R4$  register is overwritten by a loop to calculate  $x^2$  with  $R2$  register. This loop overwriting enables a program to simultaneously calculate  $x^2 + x^4$  with  $R2$  register, and two instructions, *MOVF* and *ADDWF* which calculate  $R0 \leftarrow R2 + R4 (= x^2 + x^4)$ , at the end of the program becomes unnecessary which can be removed from the program. An example of Arithmetic2 removes three instructions colored in Fig 4(b), which gets same calculation result in different calculation process. Concretely, although a program calculates  $x^y$  with combination of bit shifting and adding before removing instructions, one after evolution calculates it with only adding. In an example of Boolean2, a program into which two instructions, *BTFSZ* and *IORWF*, is added has same calculation result as before adding them, and four instructions, *MOVF*, *ANDWF*, *MOVWF*, and *IORWF*, can be removed because they become unnecessary instructions due to added two instructions.

The common feature of these evolutions is that programs in the process of evolution has either incorrect calculation result or large program size. In Arithmetic1, since a program that only overwrites a loop structure includes two adding process, it does not correctly calculate result and has low fitness. In Arithmetic2, it is necessary to remove all of three instructions simultaneously, and if at least one instruction remains in the program, it cannot also calculate correct results. In Boolean2, if both of required two instructions are added, the program can correctly calculate result, however, its size increases. This feature results to decreases the selection probability of programs that are in the evolution process, and it becomes difficult to preserve such programs in a population. Therefore, it is indispensable to maintain the diversity of programs to achieve destructive evolution.

Multi-step destructive evolution is necessary to generate programs that size is smaller than 43, 17, 23 in Arithmetic1, Arithmetic2, and Boolean2, respectively. As results of Fig 2, although such evolution can be achieved in TAGP, SGPs can achieve in few trials. Particularly, SGPs never achieve



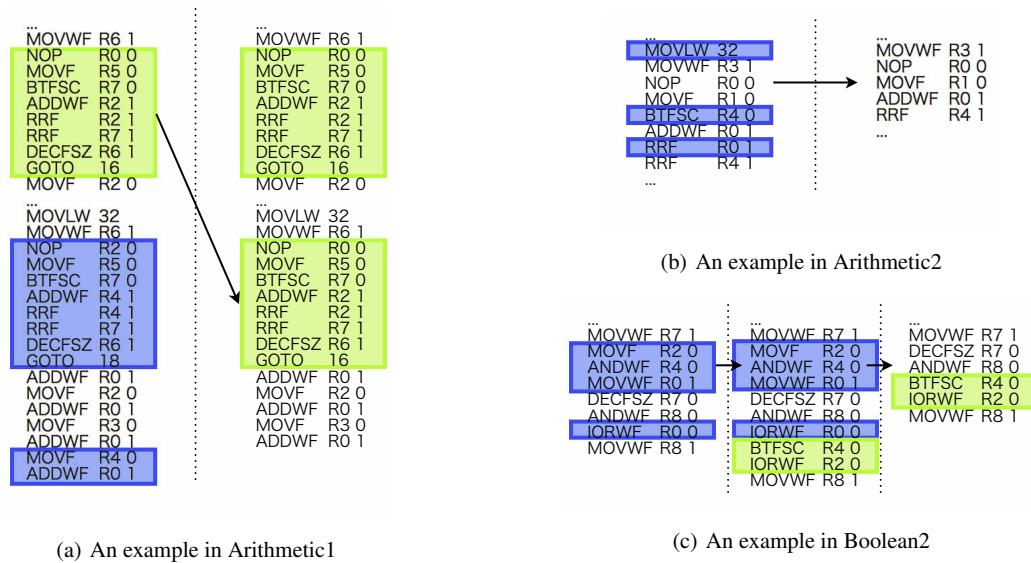


Figure 4: An example of multi-step destructive evolution

Table 7: Summary of program evolution analyses

	non-destructive	destructive
single-step	all GPs	-
multi-step	all GPs TAGP>SGPs	TAGP

this evolution in Boolean2, this is because Boolean2 requires more than three steps to generate programs that size is smaller than 23.

These results summarized in Table 7 reveal that SGPs achieve single-step non-destructive evolution which is easier than other evolutions. They, additionally, achieve multi-step non-destructive evolution in some trials. In contrast, it is revealed that TAGP cannot only achieve multi-step non-destructive evolution at a higher rate than SGPs (particularly in Boolean1), but also achieve destructive evolution which cannot be achieved SGPs. This result indicates that TAGP has higher evolution ability than SSGP and  $(\mu + \lambda)$ -GP.

### Discussion: diversity of programs

From the analyses of the program evolution, it is revealed that the diversity of programs is required to achieve complex program evolution. To confirm the diversity of programs in three GP methods, we verify the relation between the average fitness and the standard deviation of the program size in the memory/population of Arithmetic1 when a program that size is 44 and does not include unnecessary instructions is generated. Fig. 5 shows the scatter plot of the average fitness and the standard deviation of the program size for all

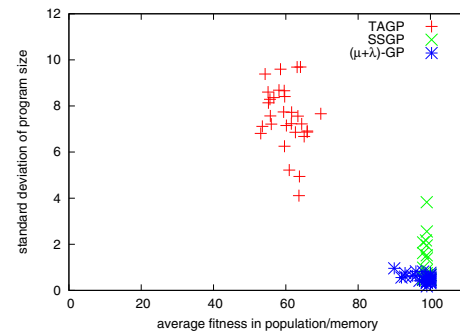


Figure 5: A scatter plot of the average fitness and the standard deviation of the program size in the memory/population of Arithmetic1

trials in Arithmetic1. In Fig. 5, the abscissa indicates the average fitness, while the ordinate indicates the standard deviation of the program size. The red points show the result of TAGP, the green points show the result of SSGP, while the blue points show the result of  $(\mu + \lambda)$ -GP. Note that the result of Arithmetic1 is only shown, but same trends are verified in other problems.

As shown in Fig. 5, it is indicated that both of SGPs have high average fitness and low standard deviation of the program size. This means that all programs in the population has very high fitness near the maximum, and also has similar program size. This result indicates that all programs are very similar to each other and the diversity of programs is very low. As mentioned in previous section, although the diversity of programs is required to generate a program that

size is less than 44 with either evolution cases, the diversities of SGPs are very low. From this result, it is revealed that it is difficult for SGPs to achieve multi-step destructive evolution because enough diversity of programs cannot be maintained.

While in TAGP, it is indicated that the average fitness is not maximum but is higher than 50, which is the half of the maximum fitness, and the standard deviation of the program size is also high. This indicates that TAGP can maintain lower fitness or larger size programs in the population, *i.e.*, several kind of programs are maintained. From this result, it is revealed that the high diversity of programs in TAGP contributes to achieve multi-step destructive evolution. This result advocates that TAGP have same feature of EAs using the asynchronous evaluation to maintain proper diversity of programs and to have high evolution ability.

### Conclusion

To investigate the evolution ability of TAGP as GP using the asynchronous evaluation, this paper compared TAGP with two simple GP methods, steady-state GP (SSGP) and GP using  $(\mu + \lambda)$ -selection  $((\mu + \lambda)$ -GP) as GP using the synchronous evaluation. Intensive comparisons among three GP methods were conducted in four computational problems to minimize the size of an actual assembly language program.

We classify the evolution processes to two categories, *non-destructive evolution* and *destructive evolution* depending on whether the evolution process affects calculation results or not. The experimental result has revealed that the following implications: (1) TAGP has higher evolution ability than SSGP and  $(\mu + \lambda)$ -GP, *i.e.*, TAGP cannot only achieve *non-destructive evolution* which is easy to be accomplished, but also achieve *destructive evolution* which cannot be achieved by SSGP and  $(\mu + \lambda)$ -GP; and (2) the diversity of the programs in TAGP can derive a high evolution ability in comparison with SSGP and  $(\mu + \lambda)$ -GP. In detail, such diversity is indispensable to *destructive evolution*.

The following issues should be pursued in the near future: (1) experiments on other problems such as classification, (2) a comparison with other GP methods, (3) an improvement of evolution ability of TAGP, and (4) a parallelization of TAGP.

### Acknowledgements

This work was supported by Grant-in-Aid for JSPS Fellows Grant Number 249376.

### References

- ATR Evolutionary Systems Department (1998). *Artificial Life and Evolutional System*. Tokyo Denki University Press.
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197.
- Epstein, J. M. and Axtell, R. L. (1996). *Growing Artificial Societies: Social Science from the Bottom Up (Complex Adaptive Systems)*. The MIT Press, 1st printing edition.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1 edition.
- Harada, T., Otani, M., Matsushima, H., Hattori, K., Sato, H., and Takadama, K. (2011). Robustness to Bit Inversion in Registers and Acceleration of Program Evolution in On-Board Computer. *Journal of Advanced Computational Intelligence and Interlligent Informatics (JACIII)*, 15(8):1175–1185.
- Harada, T., Otani, M., Matsushima, H., Hattori, K., and Takadama, K. (2010). Evolving Complex Programs in Tierra-based On-Board Computer on UNITEC-1. In *International Astronautical Congress (IAC), 2010 61st World Congress on*.
- Jong, D. and Alan, K. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, Department of Computer and Communications Sciences, University of Michigan.
- Koza, J. (1992). *Genetic Programming On the Programming of Computers by Means of Natural Selection*. MIT Press.
- Langton, C. G. (1989). *Artificial Life*. Addison-Wesley.
- Lin, L. and Gen, M. (2009). Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 13(2):157–168.
- Microchip Technology Inc. (2007). *PIC10F200/202/204/206 Data Sheet 6-Pin, 8-bit Flash Microcontrollers*. Microchip Technology Inc.
- Nonami, K. and Takadama, K. (2007). Tierra-based Space System for Robustness of Bit Inversion and Program Evolution. In *SICE, 2007 Annual Conference*, pages 1155–1160.
- Ray, T. S. (1991). An approach to the synthesis of life. *Artificial Life II*, XI:371–408.
- Reynolds, C. W. (1993). An evolved, vision-based behavioral model of coordinated group motion. In *Proc. 2nd International Conf. on Simulation of Adaptive Behavior*, pages 384–392. MIT Press.
- Storn, R. and Price, K. (1997). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. of Global Optimization*, 11(4):341–359.
- Subbu, R., Sanderson, A. C., and Bonissone, P. P. (1998). Fuzzy Logic Controlled Genetic Algorithms versus Tuned Genetic Algorithms: An Agile Manufacturing Application. *Proceeding of the 1998 IEEE international symposium on intelligent control (ISIC)*, pages 434–440.
- Yun, Y. and Gen, M. (2003). Performance Analysis of Adaptive Genetic Algorithms with Fuzzy Logic and Heuristics. *Fuzzy Optimization and Decision Making*, 2(2):161–175.
- Zhang, Q. and Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evolutionary Computation*, 11(6):712–731.

# Evolution of Spatial Pattern Formation by Autonomous Bio-Inspired Cellular Controllers

Payam Zahadat<sup>1</sup>, Thomas Schmickl<sup>1</sup> and Karl Crailsheim<sup>1</sup>

<sup>1</sup>Artificial Life Lab of the Department of Zoology, Universitätsplatz 2,  
Karl-Franzens University Graz, 8010 Graz, Austria  
payam.zahadat@uni-graz.at

## Abstract

In this paper, a gene regulatory network called FGRN (Fractal Gene Regulatory Network) and a reaction-diffusion system called AHHS (Artificial Homeostatic Hormone System) are investigated for spatial pattern formation. The two bio-inspired controllers possess similar and different features in terms of their underlying processes, structures, and communication abilities. By comparing their behaviours and capabilities in pattern formation, we provide a deeper understanding of the effects of their features. The controllers are evolved and investigated for producing various patterns in presence of different implicit positional information as well as developing a memory in order to keep the desired pattern when the positional information is eliminated. The behaviours of the controllers in each case are discussed and a preliminary test of robustness is performed. The experiments represent a positive impact of diffusion process in AHHS that is compensated by the complex structure of FGRN in producing patterns in presence of positional information and a negative effect of diffusion process in memory capability.

## Introduction

Formation of spatial patterns is a challenging subject both in biological and artificial organisms. Different forms with various levels of complexity are found everywhere in nature. One of the challenges in developmental biology is to understand the underlying processes that control the pattern formation (Jaeger and Martinez-Arias (2009)). On the other hand, from the point of view of multi-modular robotics, a proper behaviour emerges from a proper pattern of roles assigned to the modules across the body of a robot.

A problem encountering pattern formation is symmetry-breaking. In biological organisms it happens at early developmental phases. As it was suggested by Wolpert (1968) and is found in embryos, e.g. fruit fly *Drosophila melanogaster* (Driever and Nusslein-Volhard (1988); Ephrussi and Johnston (2004)), the polarization of an organism is induced by some maternal cue in the form of morphogen gradients. By using these gradients in the environment of the organism, some information is provided that is used for localization of the organism's units (cells) and participates in the process of development. The same concept is useable in artificial organisms (e.g. localization of modules in a modular robot).

For subdivision of a body using positional information, Wolpert (1968) proposed a French-flag model. The model is composed of three stripes with different colours along the body and is used by many researchers with different approaches of evolving systems (e.g., Miller (2003); Bowers (2005); Cussat-Blanc et al. (2011)).

In the field of artificial life and evolutionary computation, various models are inspired by genetic and chemical systems in biological organisms. Gene Regulatory Networks (GRNs) and reaction-diffusion models are two examples of these systems that have drawn attention in the recent years. They consist of a number of different underlying processes that control their dynamics. Although the source of inspiration and the details are different for these two models, but there are some similarities between the models. In this work, a reaction-diffusion model and a GRN model are investigated in the context of pattern formation.

GRNs are inspired by internal interactions between genes and proteins in cells. Various models of computational GRNs have been defined and investigated from different perspectives, e.g. studying dynamics (Banzhaf (2003)), applying for morphology development (Eggenberger (1997); Roggen and Federici (2004)), developing both morphology and controller of robots (Bongard and Pfeifer (2001)).

Fractal Gene Regulatory Network (FGRN) (Bentley (2004b)) is an example of GRN models. It is originally designed as a single-unit of control and successfully implemented for different tasks, i.e., controlling conventional robots (Bentley (2004a)) and pole-balancing (Krohn and Gorse (2010)). Since no explicit communication mechanism between different units is defined in FGRN, environmental feedback is used to coordinate modules in multi-modular robotic applications of FGRN (Zahadat et al. (2010, 2012)).

Reaction-diffusion models are inspired by intracellular signaling in biological organisms. The models contain both a process of local reaction between substances and diffusion of substances across the organism. Artificial Homeostatic Hormone System (AHHS) is an example of these models which is originally introduced in Schmickl and Crailsheim (2009) and has been used successfully in robotic applica-

tions for both single and multi-modular robots (Stradner et al. (2009); Schmickl et al. (2010); Hamann et al. (2010)).

In this paper, FGRN and AHHS are evolved and investigated for generating target patterns with fixed maternal morphogen gradients as well as generating a memory such that the target pattern is preserved after elimination of the maternal gradients. Behaviours of different evolved solutions are discussed and a sample evolved controller is tested for its reaction against an instant reset in a single unit in order to have an evaluation of robustness of the produced pattern.

While the two systems are similar in terms of having mechanisms to produce various mappings between input and output as well as forming internal feedback loops, they are different in their complexity and details of mechanisms, structure, and communication abilities. For instance, FGRN model provides more complicated interaction network between local substances in comparison with AHHS. A prominent difference between the two models is the lack of intra-unit communication in FGRN. Due to that, the spatial patterns generated by FGRN lay solely on the maternal gradients and internal interactions of each particular unit. On the other hand, in AHHS, along with the maternal gradients and internal interactions, the pattern formation can benefit from the diffusion of substances over the units.

In this work, in addition to FGRN and AHHS with their standard underlying processes, a diffusion-free version of AHHS has been also implemented in order to investigate the importance of diffusion in the observed differences between the behaviours of FGRN and AHHS.

### Short Summary of FGRN

FGRN (Bentley (2004b)) is a GRN model that uses an abstract model of proteins, called fractal proteins, as the means of interaction between genes. These means of interactions are encoded in the genome and evolved by a version of Genetic Algorithm (GA) (see Bentley (2004b) for details).

The genome consists of a number of genes and parameters. Every gene in the system belongs to a type of genes: input genes, output genes, regulatory genes and receptor genes. Input, regulatory and receptor genes encode corresponding fractal proteins. A fractal protein has a shape and a concentration level. The shape is encoded in a gene by three real values. These values determine a square window on Mandelbrot fractal set. A protein's concentration level is a variable value. The changes in concentration level is controlled by the other proteins' concentration levels, shapes of the fractal proteins, and other parameters of the genome. To every sensory input into an FGRN system, a set of input proteins are associated. The input value determines the concentration levels of the corresponding input proteins and consequently participate in driving the dynamics of the system. Receptor proteins act as filters over inputs by manipulating shapes of input proteins. Regulatory proteins participate in driving the internal dynamics. Their concentration levels are

both controlled by and also participate in controlling the dynamics of the system. In fact, they make regulatory connections in the network of proteins and are potentially capable of establishing recurrent loops and act as a sort of memory in the system. Output genes determine the influence of the concentration levels of the proteins on the output of the system. (For a detailed introduction of FGRN see Bentley (2004b))

FGRN can be seen as several systems of Difference Equations ( $O\Delta E$ s) where each  $O\Delta E$  system controls the internal dynamics over time in a particular part of the state space. Concentration levels of proteins are state variables of the system. When the value of a state variable changes from a positive value to zero or vice versa, the system switches between different parts of the state space and its behaviour changes due to activation of a different system of  $O\Delta E$ s (For a detailed description of this representation of FGRN see Zahadat and Støy (2012)). Table 1 demonstrates an example FGRN represented as conditional sets of  $O\Delta E$ s.

Table 1: An example of a simple FGRN as several conditional sets of  $O\Delta E$ s. The state-space of this system is divided into four parts.  $S$  represents the set of proteins with positive concentration levels indicating a division of the state-space.  $P_1$  and  $P_2$  correspond to an input and a regulatory protein respectively and  $p_1$  and  $p_2$  are their corresponding concentration levels.  $out$  is the output of the system.

condition	equation set
if $S = \{P_1, P_2\}$	$p_2 \leftarrow 0.8p_2 - (0.2p_1 + 0.5p_2)$ $\quad \times \tanh(0.6p_1 + 1.5p_2 - 3.6) - 0.2$ $out \leftarrow 0.15p_1 + 0.2p_2 + 4$
if $S = \{P_1\}$	$p_2 \leftarrow 0.4p_1 + 0.5$ $out \leftarrow 0.32p_1 + 2$
if $S = \{P_2\}$	$p_2 \leftarrow 0.8p_2 - 0.25p_2$ $\quad \times \tanh(0.5p_2 - 0.4) - 0.2$ $out \leftarrow 0$
if $S = \{\}$	$p_2 \leftarrow 0.4$ $out \leftarrow 0.25$

Due to the lack of any intra-unit communication in FGRN, symmetry breaking and differentiation of units is achievable by providing different inputs for different units.

### Short Summary of AHHS

AHHS (Artificial Homeostatic Hormone System) (Schmickl and Crailsheim (2009)) is a reaction-diffusion-based system inspired by Turing process (Turing (1952)) that describes processes of natural pattern formation and growth.

An AHHS is defined by a set of artificial hormones and a set of rules. The rules define how sensory input and hormone concentrations participate in changing the concentrations and outputs of the system. Both hormones and rules are evolved by using a standard real-valued GA.

An AHHS can be represented as a dynamical system consisting of several state variables (hormone concentrations) and a system of  $O\Delta E$  that governs their dynamics. The key feature of AHHS is how the parameters of this  $O\Delta E$  are encoded and determined. Concentrations of the hormones are



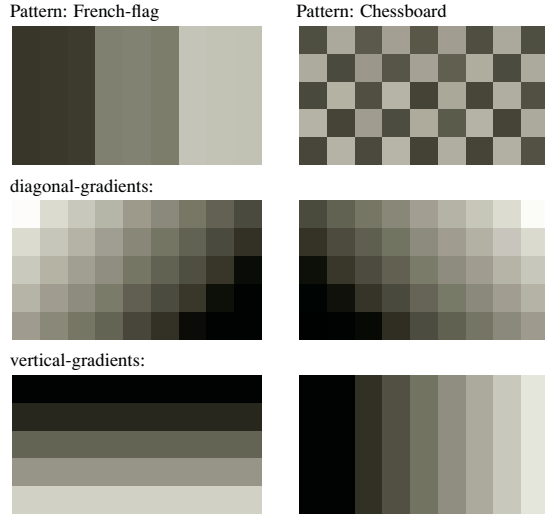


Figure 1: Target patterns (first row), diagonal gradients (second row), and vertical gradients (third row).

allowed to increase independently by a base production rate and are also subject to a certain decay.

The dynamics of hormone concentration  $H$  at time  $t$  is defined for hormone  $h$  as follows:

$$\frac{\Delta H_h}{\Delta t} = \alpha_h + D_h \nabla^2 H_h(t) - \mu_h H_h(t) + \sum_i \mathcal{L}_i(t), \quad (1)$$

where  $\alpha_h$ ,  $D_h$ , and  $\mu_h$  are base production rate, diffusion rate, and decay rate of hormone  $h$  respectively.  $\mathcal{L}_i(t)$  is the influence of rule  $i$  on hormone  $h$  and is defined as:

$$\mathcal{L}_i(t) = \theta(H_k(t))(H_k(t)\lambda_i + \kappa_i), \quad (2)$$

$$\theta(x) = \begin{cases} 1 & \text{if } \min_i < x < \max_i \\ 0 & \text{else} \end{cases}, \quad (3)$$

where  $\lambda_i$ ,  $\kappa_i$ ,  $\min_i$  and  $\max_i$  are parameters of the rule.  $\theta(H_k(t))$  determines if the rule is triggered or not. If the rule is triggered, the concentration of hormone  $h$  changes linearly based on concentration of hormone  $k$ .

In the current implementation, sensory inputs are scaled in the range of hormone concentrations  $[0,1]$  and directly set to the concentration level of particular hormones. In the same way, concentration level of a particular hormone is considered output of the AHHS unit.

## Experiments

In this work, the systems are evolved to produce patterns in a  $5 \times 9$  rectangular grid. Every cell of the grid has a colour which is determined by the output of its controller. The controllers are genetically identical all over the grid. Two maternal gradients are provided over the grid in every experiment.

Two types of target patterns and two types of maternal gradients are considered with different degrees of difficulty (see Figure 1 for both target patterns and maternal gradients). For the FGRN controllers, the two maternal gradients enter a unit in form of two sensory inputs that in turn influence the concentration levels of input proteins of the unit. In AHHS system, the values of the maternal gradients directly set the concentration levels of two particular hormones. The output from each unit (either a FGRN or AHHS controller) is mapped into one of the three predefined colours regardless of the number of colours in a particular target pattern.

The controllers are evolved for the following tasks:

- Since French-flag pattern as suggested in Wolpert (1968) is a benchmark in evolving for pattern formation, in the first task, the target pattern is a French-flag with vertical maternal gradients (third row of Figure 1). In this case we suspect that the controllers make a direct mapping between one of the maternal gradients and the output.
- In the second task, the target pattern is again a French-flag but this time the maternal gradients are diagonal, as in Cussat-Blanc et al. (2011) (second row of Figure 1).
- The third task aims at producing a chessboard pattern with diagonal maternal gradients. In the first three tasks, the system runs for 100 time-steps while the gradients are stable during the run-time.
- In the forth task, the controllers are supposed to produce a sort of memory. The gradients are presented for the first few time-steps when the target pattern is generated. Then the gradients are removed while the target pattern is expected to be preserved by the system. The simplest combination of target pattern and gradients (French-flag with vertical gradients) is chosen for this task in order to keep the focus on the memory capability.

## Evolving for the target patterns

Populations of 50 random individuals are evolved for each task for FGRN, standard AHHS, and diffusion-free AHHS controllers. Every experiment is repeated for 10 independent runs with 1500 generations. Table 2 represents the controller and evolutionary settings.

Table 2: Controller and evolutionary settings

FGRN:

population-size	50	#generations	1500	#recomb.	0.4
mut. prob.	1	#receptors	2	#inputs	2
#regulatory	2	#outputs	2		

AHHS:

population-size	50	#generations	1500	recomb. prob.	0.01
mut. prob.	0.4	#hormone	6	#rules	30

In the first three tasks, fitness is defined as the number of correct coloured cells in the final pattern. In the case of evolving for memory, two fitness factors are required. A fitness factor is required to direct the evolution towards generating target pattern and the second factor is needed for evolution of the memorizing ability. Considering these factors, fitness is defined as a combination of the number of correct colours in the last time step with maternal gradient (time-step 10) and the average number of correct colours in all the next time steps:

$$fitness = 3 \times C_{T_p} + \frac{1}{T - T_p} \sum_{t=T_p+1}^T C_t$$

where  $T$  is the number of time-steps of the experiment and in the first  $T_p$  time-steps the gradient is present,  $C_{T_p}$  is the number of correct colours in time-step  $T_p$ , and  $C_t$  is the number of correct colours in time-step  $t$ .

Comparisons between FGRN, standard AHHS, and diffusion-free AHHS controllers for all the tasks are represented in Figure 2. The figure represents that FGRN and AHHS are evolvable to produce the pattern for all the tasks while none of the runs of diffusion-free AHHS produces a perfect French-flag and chessboard with diagonal gradient. Figure 3 demonstrates the median fitness-progression of the controllers over generations. The figure represents that the French-flag is easy-to-produce for all the controllers when the gradients are vertical. In the case of diagonal gradients for both French-flag and chessboard, diffusion-free AHHS has the lowest fitness indicating that diffusion has a positive effect for AHHS in these tasks. In the case of evolution for memory, diffusion-free AHHS makes higher fitness than standard AHHS while FGRN represents the highest value. It implies a negative effect for diffusion in keeping a pattern without external clue of maternal gradients.

### Looking at the behaviours

In order to make an impression of the solutions of each controller type, we will have a look at the behaviours of the evolved controllers in the following sections. Representative examples of different observed behaviours in each task are displayed in Figure 4. Each curve in the figure represents the development of fitness achieved by an example controller over time (Note that the fitness in the last time-step is considered the actual fitness of the controller).

**French-flag with stable vertical gradients** All the three types of controllers are able to produce perfect target pattern. The difference between the number of successful runs is not statistically significant (Figure 2).

In FGRN, 9 runs out of 10 produce target pattern perfectly. In six runs out of 10, the pattern is perfectly generated from the first time-step that represents that static controllers are found that simply threshold the maternal gradients to produce the output that is mapped to the respective

colours. The other three runs reach the perfect pattern after less than 20 time-steps and then the pattern stays stable.

In AHHS, 9 out of 10 runs produce perfect pattern. In four runs, the output oscillates between different patterns such that in time-step of observation (time-step 100) the perfect pattern is represented. In the other five runs, fitness increases gradually and perfect pattern is produced after several steps and stays stable until time-step 100. Generation of perfect pattern is slower than similar cases in FGRN.

In diffusion-free AHHS, 9 runs out of 10 generate the pattern. Five runs generate oscillatory patterns such that the target pattern is represented in time-step 100. In the other four runs, the pattern is slowly produced during time and stays stable. The increase in fitness is again slower than FGRN but no significant difference with standard AHHS is observed.

**French-flag with stable diagonal gradients** In FGRN, four runs out of 10 reach the perfect pattern. Three of them make the correct pattern from the first step indicating thresholding of the maternal gradients. The other run, makes the pattern after about 10 steps. The unsuccessful runs mainly produce changing patterns with chaotic oscillation in the fitness curve, although a static pattern is observed in one run.

In AHHS, four runs produce the perfect target pattern. All the four successful runs generate the correct pattern very slowly and gradually in last time-steps. In all the other runs the fitness gradually increases during time although short decreases are also not impossible. No oscillation is observed.

In diffusion-free AHHS, none of the runs are successful to produce perfect target pattern. Changing patterns with both chaotic and ordered oscillations and also patterns with gradual increase in the fitness are observed.

**Chessboard with stable diagonal gradients** In FGRN, three runs reach the perfect target pattern and two other runs are correct except for one cell of the grid. Although we suspected that the success of FGRN in this case might be a result of precise thresholding but the patterns oscillate over time such that the perfect pattern is presented in the last time-step. In fact all the ten runs make oscillatory patterns mostly with big differences from one step to the next.

In AHHS, one run reaches the perfect pattern. This pattern is not stable and oscillates between the correct and the inverse patterns such that in the last time-step the perfect pattern is presented. The oscillation is more ordered comparing the FGRN runs. Most of the other runs produce oscillating patterns but stable patterns are also observed.

In diffusion-free AHHS, no run produces the perfect pattern. The runs generate different patterns and although some stable patterns are generated but most of the runs make oscillatory patterns. In comparison with AHHS, ordered oscillations with high frequencies are not common and oscillations are more chaotic and in comparison with FGRN the oscillations have lower frequencies.

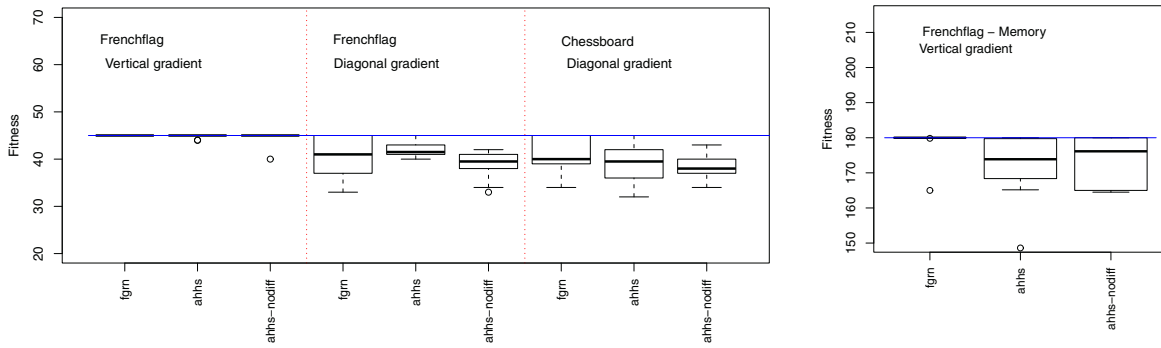


Figure 2: Fitnesses of the best individuals in the last evolutionary generation for FGRN, AHHS, and diffusion-free AHHS controllers for the four tasks. Box-plots indicate median and quartiles, whiskers indicate minimum and maximum, circles indicate outliers (values are collected from 10 independent runs).

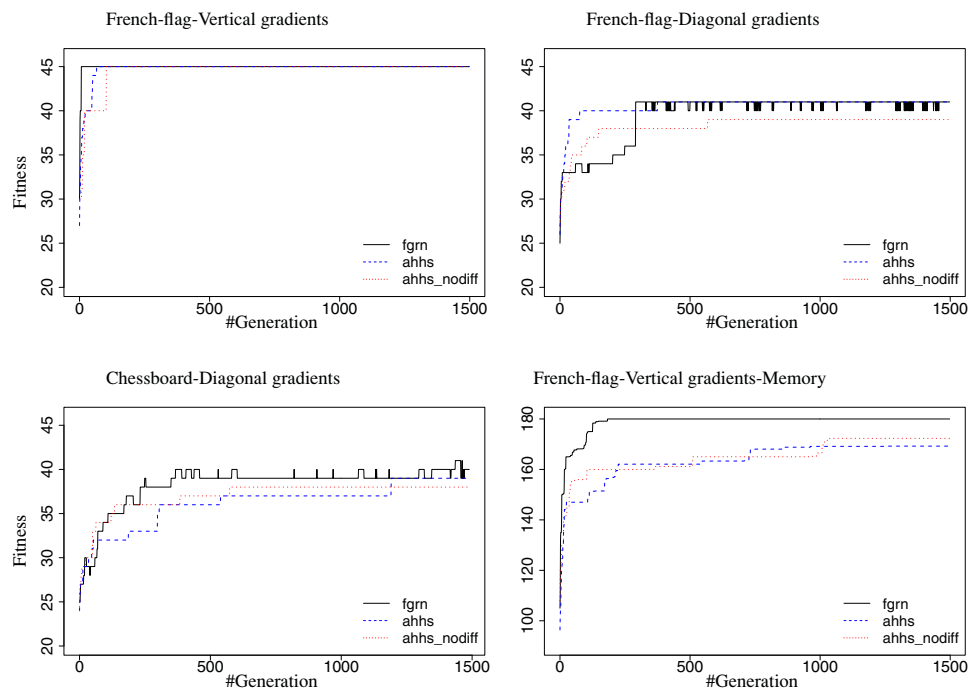


Figure 3: Comparison of fitness trajectories for FGRN, AHHS, and diffusion-free AHHS controllers in the four tasks. The values are medians of the best fitnesses in the 10 independent evolutionary runs. (y-axis starts from 20 due to the space-limitation and since there is no data below 20.)

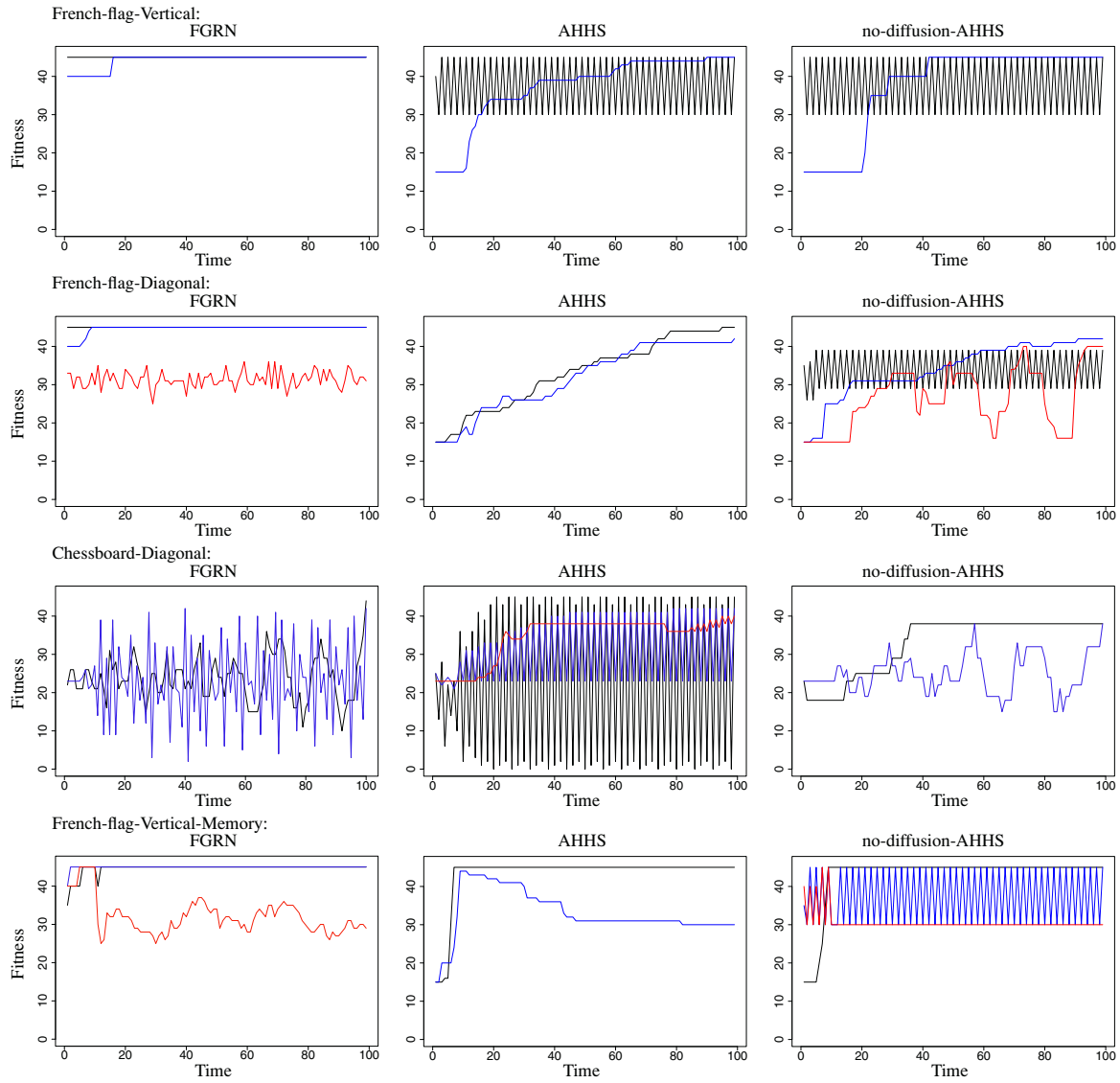


Figure 4: Fitness development over time for representative examples of the evolved solutions for the three controller types.

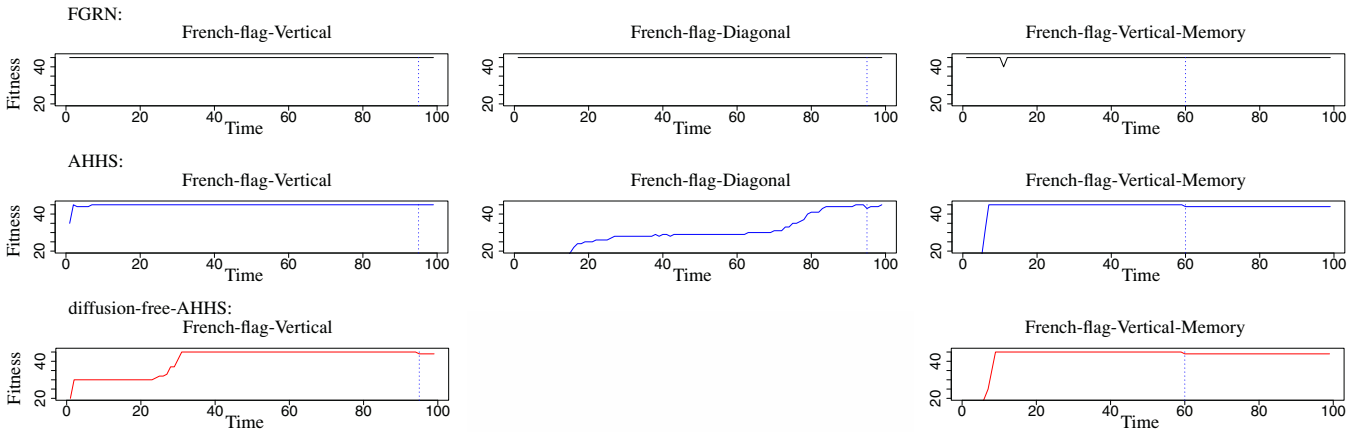


Figure 5: Behaviour of example controllers when all the state variables of the cell in the middle of the grid is reset to zero in a particular time-step that is represented by the vertical dashed-lines. The diagrams are squeezed due to space-limitation.



**Memory in French-flag with vertical gradients** In FGRN, four runs are successful in producing the pattern at time-step 10 (last time-step with gradients) and keeping it for the next 90 time-steps (with no gradient). In three other runs, only for one step after vanishing the gradient, the pattern is disturbed and then is regenerated. In the other three runs, the target pattern is produced in time-step 10 but then it changes and the fitness oscillates chaotically or ordered.

In AHHS, in two runs the pattern is produced until time-step 10 and is kept until the end. In one run, the pattern is represented in time-step 10, but when the gradients are vanished the pattern slightly deviates from target for few time steps but again is generated and kept until the end. In another run, the perfect pattern in time-step 10 is kept for about 60 time-steps and then deviates from target in a single cell of the grid until the end. In all the rest except one, the target pattern is represented in time-step 10, but then it deviates from target when the gradients are vanished.

In diffusion-free AHHS, five runs produce and keep the perfect target pattern from time step 10 to 100. In one run, the produced pattern in time-step 10 is perfect, then it changes in the next time step when the gradients are vanished but it is produced again after one step and is kept until the end. This is the effect that is also observed in FGRN. In one run, the pattern is switched between the target and a simpler pattern in every step. The other three runs produce the target pattern in time-step 10 and then the pattern changes to a simpler pattern and stays stable until the end.

### Robustness in example controllers

In the last experiment we aim to evaluate robustness of the controllers by disturbing the internal variables of a cell in the grid in order to see if the correct pattern is regenerated after few time-steps. Experiments are performed with the French-flag pattern in the three cases of stable vertical gradients, stable diagonal gradients, and memory with vertical gradients, for all the controller types. In each case, we chose a controller from the previous experiments that produces the target pattern long before the end of evaluation period. Then the controller is used to produce the pattern and in a particular time-step all the dynamic values (hormones/proteins) of the controller in the middle of the grid are set to zero. The fitness of the system at the end of the evaluation period is then calculated that represents whether the pattern is regenerated or not. The results are represented in Figure 5. Since diffusion-free AHHS did not evolve for French-flag pattern with diagonal gradients in any of the runs, it is omitted in the figure. The figure demonstrates no change in any of the patterns produced by FGRN controllers indicating robustness of the controller against the reset. In AHHS controller, for the stable vertical gradients the pattern does not change after reset. In the stable diagonal gradients, the pattern changes (fitness decreases) but it is reproduced after few time-steps. In the case of memory as well as the tasks with diffusion-free

AHHS, the patterns do not regenerate after reset.

## Conclusions

FGRN (Fractal Gene Regulatory Network), standard AHHS (Artificial Homeostatic Hormone System), and a diffusion-free AHHS are evolved and investigated for their capability in pattern formation in presence of maternal gradients.

FGRN is computationally expensive due to encoding of proteins as the intermediate substances that establish interaction connections between genes. By this indirect encoding, a potentially complex interaction network of genes is formed. FGRN defines no communication mechanism between units and all the dynamics of the system are based on the internal interaction network. On the other hand, AHHS is a reaction-diffusion-based model and employs diffusion as a means of communication between units. In AHHS, the interaction connections between hormones are directly encoded which provides a simpler interaction network inside a unit, in comparison with FGRN. The dynamics of the system is based on both the internal interaction network and the diffusion mechanism.

In this work, three sets of evolutionary experiments with different combinations of target patterns and stable gradients are performed. In addition, a memory test experiment is performed where the maternal gradients are provided in the first time-steps and then they are vanished from the environment. The system is expected to produce the target pattern in presence of maternal gradients and keep it intact after elimination of the gradients.

In all the three experiments with stable maternal gradients, both FGRN and standard AHHS are successfully evolved for the perfect solution although the rates of success are different. It has to be mentioned that in a previous work by Cussat-Blanc et al. (2011), successful evolution of another GRN model (Banzhaf (2003)) in the case of French-flag with diagonal gradients was reported where the result of a single evolutionary run was presented. Diffusion-free AHHS is evolved successfully for producing the French-flag pattern in presence of vertical maternal gradients but it did not find the solutions in more complicated tasks of producing chessboard pattern and French-flag with diagonal maternal gradients. It implies that diffusion is an important part of an AHHS system. On the other hand, despite of importance of diffusion in AHHS, FGRN is also successful in all the tasks. It indicates that in principle the investigated patterns are producible even without diffusion. The more complicated nature of FGRN system enables it to be evolved for the proper patterns although there is no communication mechanism implemented in FGRN. In addition, observing the evolved solutions for French-flags (both gradient types) represents that FGRN can produce the pattern in one time-step while AHHS always needs time to build it up. This is also due to the complex structure of FGRN in terms of  $O\Delta Es$ .

All the three types of controllers were successfully evolved for the memory experiment. The rate of success (number of perfect evolutionary runs) was lower for standard AHHS in comparison with diffusion-free AHHS and FGRN. It can lead to the conclusion that diffusion has a negative effect in keeping the memory. This effect is intuitively expected, since diffusion tends to flatten the pattern and having that in the system requires a compensation mechanism, e.g. elaborating internal feedback loops.

A preliminary experiment for evaluating robustness of the systems has been also performed in this work and represented the highest robustness for the solutions generated by FGRN system. In the future, controllers with different subsets of internal processes will be evolved for series of spatial patterns with increasing levels of complexity and the effects of the internal processes will be investigated in details.

### Acknowledgements

This work is supported by: EU-ICT project ‘CoCoRo’, no. 270382; EU-IST-FET project ‘SYMBRION’, no. 216342; EU-ICT project ‘REPLICATOR’, no. 216240; Austrian Federal Ministry of Science and Research (BM.W F); EU-ICT project ‘ASSISI.bf’, no. 601074.

### References

- Banzhaf, W. (2003). Artificial regulatory networks and genetic programming. In *Genetic Programming Theory and Practice*, pages 43–62. Kluwer.
- Bentley, P. J. (2004a). Adaptive fractal gene regulatory networks for robot control. In *Workshop on Regeneration and Learning in Developmental Systems in the Genetic and Evolutionary Computation Conference (GECCO 2004)*.
- Bentley, P. J. (2004b). Fractal proteins. *J Genet. Program Evol. Mach*, 5:71–101.
- Bongard, J. C. and Pfeifer, R. (2001). Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny. In *Proceedings of The Genetic and Evolutionary Computation Conference*, pages 829–836. Morgan Kaufmann.
- Bowers, C. P. (2005). Simulating evolution with a computational model of embryogeny: Obtaining robustness from evolved individuals. In *In Advances in Artificial Life, Proceeding of the 8th European Conference on Artificial Life: ECAL 2005 (2005)*, pages 149–158.
- Cussat-Blanc, S., Bredeche, N., Luga, H., Duthen, Y., and Schoenauer, M. (2011). Artificial Gene Regulatory Network and Spatial Computation: A Case Study. In *European Conference on Artificial Life*, Paris, France.
- Driever, W. and Nusslein-Volhard, C. (1988). The bicoid protein determines position in the drosophila embryo in a concentration-dependent manner. *Cell*, 54(1):95–104.
- Eggenberger, P. (1997). Evolving morphologies of simulated 3d organisms based on differential gene expression. In *Proceedings of the fourth european conference on Artificial Life*, pages 205–213. MIT Press.
- Ephrussi, A. and Johnston, D. S. (2004). Seeing is believing - the bicoid morphogen gradient matures. *Cell*, 116(2):143–152.
- Hamann, H., Stradner, J., Schmickl, T., and Crailsheim, K. (2010). Artificial hormone reaction networks: Towards higher evolvability in evolutionary multi-modular robotics. *Proc. of the ALife XII Conference*, pages 773–780. MIT Press.
- Jaeger, J. and Martinez-Arias, A. (2009). Getting the Measure of Positional Information. *PLoS Biology*, 7(3):e81+.
- Krohn, J. and Gorse, D. (2010). Fractal gene regulatory networks for control of nonlinear systems. In *Lecture Notes in Computer Science*, volume 6239, pages 209–218. Springer-Verlag.
- Miller, J. F. (2003). Evolving developmental programs for adaptation, morphogenesis, and self-repair. In *Advances in Artificial Life. 7th European Conference on Artificial Life*, volume 2801 of *Lecture Notes in Artificial Intelligence*, pages 256–265, Dortmund, Germany. Springer.
- Roggen, D. and Federici, D. (2004). Multi-cellular development: is there scalability and robustness to gain? In *Parallel Problem Solving from Nature 8 (PPSN'2004)*, pages 391–400.
- Schmickl, T. and Crailsheim, K. (2009). Modelling a hormone-based robot controller. In *MATHMOD 2009 - 6th Vienna International Conference on Mathematical Modelling*.
- Schmickl, T., Hamann, H., Stradner, J., Mayet, R., and Crailsheim, K. (2010). Complex taxis-behaviour in a novel bio-inspired robot controller. In *Proc. of the ALife XII Conference*, pages 648–655. MIT Press.
- Stradner, J., Hamann, H., Schmickl, T., and Crailsheim, K. (2009). Analysis and implementation of an artificial homeostatic hormone system: A first case study in robotic hardware. In *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'09)*, pages 595–600. IEEE Press.
- Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, B237(641):37–72.
- Wolpert, L. (1968). The French Flag problem: A contribution to the discussion on pattern development and regulation. In Waddington, C. H., editor, *The Origin of Life: Toward a Theoretical Biology*, pages 125–133.
- Zahadat, P., Christensen, D., Katebi, S., and Stoy, K. (2010). Sensor-coupled fractal gene regulatory networks for locomotion control of a modular snake robot. In *Proceedings of the 10th Int. Symposium on Distributed Autonomous Robotic Systems*, pages 517–530.
- Zahadat, P., Schmickl, T., and Crailsheim, K. (2012). Evolving reactive controller for a modular robot: Benefits of the property of state-switching in fractal gene regulatory networks. In *Lecture Notes in Computer Science*, pages 209–218. Springer Berlin Heidelberg.
- Zahadat, P. and Støy, K. (2012). An alternative representation of fractal gene regulatory networks facilitating analysis and interpretation. *Annals of Mathematics and Artificial Intelligence*, 65(4):285–316.

# Multiplicity of Interpretation in an Asynchronous Updating Rule: Emergence of Collective Cognition

Takayuki Niizato<sup>1</sup>

<sup>1</sup>Faculty of Engineering, Information and Systems, Tsukuba University, Japan  
t.niizato@yahoo.co.jp

## Abstract

Many kinds of interactions among individuals construct collective animal behavior, but how to apply this multiplicity of interactions is often unclear when constructing models. We propose multiplicity of interaction in a simple model constructed from three factors: asynchronous updating, learning site patterns, and agent anticipation. We found that the first two contribute to an efficient searching strategy, and that adding agent anticipation enables sign making (avoidance) in heterogeneous environments. Our model surprisingly suggests that searching strategies and territorial behavior such as boundary marking — seemingly contradicting behaviors — emerge from two aspects of our simple interaction rule. We discuss the possibility of collective cognition in animals when heterogeneous environments change. Our study suggests that multiplicity of interaction in asynchronous updating is very important for understanding many aspects of emergent collective behavior in animals.

## Introduction

Collective animal behavior results from many kinds of interactions among individuals, and interactions vary flexibly according to the situation encountered by individuals (Couzin, 2009; Sumpter, 2006). In self-organization models, behaviors can emerge by tuning some parameters (Couzin, 2009; Haken, 1983; Sumpter, 2006). However, in most cases these various behaviors are monotonic. For example, flocking behavior can form swarms, tori, or polarized groupings by tuning interaction ranges. This is monotonic behavior because it is a variation of a single aspect, in this case alignment (Couzin, 2009). Self-organization has been discussed from many aspects, but it is increasingly important to consider theoretical aspects of the essential diversity of behavior, which sometime seems mutually irrelevant or even contradictory. The emergence of various behaviors in self-organization traces back to the concept of the “subsumption architecture” proposed by Brooks /citepBrooks. Subsumption architecture can be summarized in two aspects, namely, that agents (or robots) never need representations and that the behavior of subjects in a lower layer such as object avoidance finally emerges in higher-layer behavior such as environment searching (Brooks, 1990). Most importantly,

multiple simple behaviors ultimately form qualitatively different collective behavior. To account for the above, we propose a new model of self-organization in collective animal behavior using an asynchronous updating method. In our model, each agent uses scent marking at each site passed. Scent markings are often observed in real systems, such as ants, wolves, and other animals (Cornforth et al., 2005; Giuggioli et al., 2011; Lewis and Murray, 1993).

Although there is a wide range of scent marking behavior in actual phenomena, we can consider their common properties, namely that scent marking gives collective information about where the animal lives (Giuggioli et al., 2011; Lewis and Murray, 1993). Animals thus demonstrate collective decision-making using this information. However, information about locations in scent marking inevitably has a multiplicity of meanings, because the agent’s interpretation of the information would rely heavily on the situation. In our model, asynchronous agent actions give many interpretations, or multiplicities, to this scent marking. When each agent interacts with neighbor sites of locations and other agents, our model divides asynchronous updating of agent states into an active phase and a passive phase. In the passive phase, agents only memorize their environment and obey stochastic rules. In the active phase, agents use the memorized state against their environment. As we will show, these two interaction phases, induced by asynchronous updating, exhibit non-trivial collective searching behavior compared with the normal model. The remainder of this paper is organized as follows: In section II, we describe the algorithm of our asynchronous learning model (ALM) and present the effect of two phases of interaction for collective searching behavior. In section III, we introduce “anticipation” for each agent in ALM to develop an asynchronous learning model with anticipation (ALMA). This anticipation incorporates past and future information with current information when an agent decides its next action. We show the anticipation of each agent contribute to sign making, an avoidance, for heterogeneous environments. Finally, we discuss the possibility of collective cognition for changing environments.

## Result

### Asynchronous Learning Model (ALM)

Our ALM is mainly divided into two parts, site pattern learning induced by asynchronous updates and asynchronous updating with anticipation. These two parts can be considered as one, but division clarifies the effect of these two factors. We define the neighborhood of each agent as a Moore neighborhood, meaning there are eight sites around an agent of interest. There are three site patterns: occupied, scent marked, and vacant. There are therefore  $3^8 - 1 = 6560$  neighborhood patterns.

In this section, we examine the effect of pattern learning using asynchronous updating. First, we divide the method of agent interaction into an active phase and a passive phase. Using asynchronous updating means there are two neighbor site patterns where each agent interacts with its environment (Figure 1). One is the case of no other agent around the agent whose turn it is. This case would be common in early turns, or in a low-density environment. The other case is that there are agents at the neighbor sites of the agent whose turn it is. This case is common in late turns, or in high-density environments. These are called active and passive phases, respectively. These two interaction phases have an important place in our agent interaction model.

We next assign different roles to these two phases. In the passive phase, some agents have already occupied the agent's neighbors on its turn. Assuming the agent cannot move to these occupied sites, it must select from the remaining unoccupied sites, which correspond to vacant sites or scent marked sites. Unoccupied sites are selected stochastically. Our model assigns a high selection probability to scent marked sites and a low selection probability to vacant sites. The ratio between these two probabilities,  $\text{Prob}(\text{Vacant})/\text{Prob}(\text{Scent})$ , is represented by a parameter  $\mu$  ( $0 < \mu \leq 1$ ). We will see that a low value for  $\mu$  results in agents aggregating through the use of their scent marking. Furthermore, each agent learns (or memorizes) the pattern of neighbor sites (Figure 1, Right).

Next, We set properties in the active phase. Agents used stored information corresponding to the current pattern of neighbor sites to the pattern memorized in the passive phase. The current site pattern is therefore replaced with another site pattern, which almost always includes some occupied (or block) sites. Each agent uses this stored information and stochastically selects one site from among the sites that can be moved to. Using stored information thus affects agent interpretation of the environment.

We divide the roles for the active and passive phases, respectively, as using stored information and learning the site pattern. Because asynchronous movement can create timing conflicts, there is always room for multiple interpretations of a given site pattern in one time interval. Note that the asynchronous update is not a random order scheme, but a density order scheme; high-density agent neighborhoods

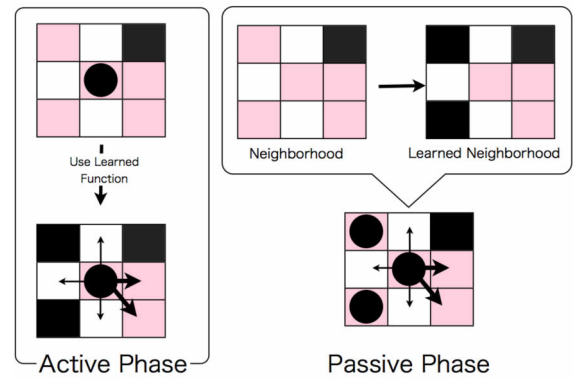


Figure 1: A sketch of the algorithm. Black circles correspond to agents. Each site color shows its state. White corresponds to a void site, pink corresponds to a scent marking site, and black corresponds to an occupied site. Each agent interacts with its neighborhood in an active phase or passive phase. The size of arrows represents the degree of the weight.

tend to be selected first, and agents with low density move last. We count as one step all agents being updated using density-dependent asynchronicity.

### Searching Strategy in ALM

Now we examine the effect of multiple site pattern interpretations induced by the asynchronous updating method. To clarify this effect, we introduce a control model constructed using asynchronous updating without site pattern learning in the active phase. In other words, there is no conflict regarding interpretation of the site pattern between the active phase and the passive phase. The parameter  $\mu = 0.001$  is the same as in the control model and the number of agents is 150.

Figure 2 shows a distribution of agents on a  $35 \times 35$  grid for our model and its control. Each color corresponds to the state of the sites. Red corresponds to an agent, pink corresponds to a scent marker, black corresponds to a block (such as an occupied site) and white corresponds to a vacant site. As compared with ALM, agents in the control model clearly form a small aggregation, meaning they are connected by their scent markers. Once agents in the control model form a local group, they hardly move.

In contrast, asynchronously updated agents with learning never broadly cover the space, they instead form aggregations, suggesting that agents with learning connect with each other to effectively search over the entire space, unlike the control model (ALM without learning). Scent markers were originally attractive signs, as we observe in the control model. The asynchronous update with site pattern learning shows an additional ability, collective searching behavior. In fact, Figure 3 shows that mean cover rate of scent marking



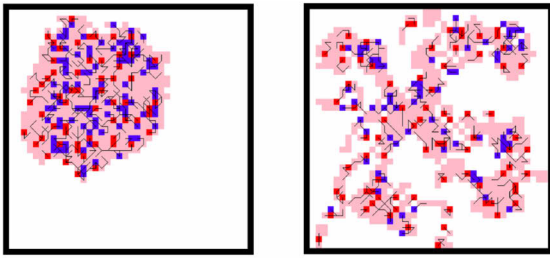


Figure 2: Agent distribution on a  $35 \times 35$  grid. The left figure shows the control model, and the right figure shows ALM. Agent colors correspond to interactions; red agents are in the active phase, and blue agents are in the passive phase. The black tail with each agent represents a trajectory in a few steps.

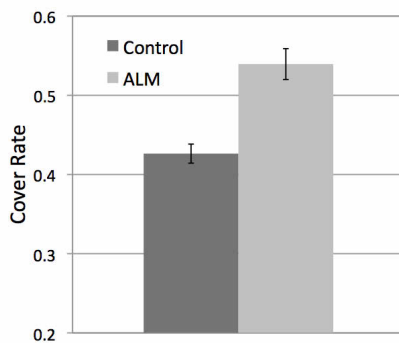


Figure 3: Mean cover rate of scent marking. The ALM value (light gray) is higher than the control model value (gray).

area is larger than the control. This result suggest agent in ALM do more efficient searching than the control model despite the almost same mechanism.

This additional ability is because of the repulsion effect against dense spaces when an agent uses site pattern learning in the active phase. No agent keeps staying in the same group, due to this repulsion effect. In other words, the conflict (or difference) between the active phase and the passive phase allows multiple interpretations of the environment for each agent.

From an ecological perspective, this difference is important. Generally speaking, animals live in groups, which has advantages in reproduction and vigilance against predators (Jackson, 2006; Johnson et al., 2002; Parrish, 1999). Forming groups is therefore an essential problem for living animals. Group living can also be an issue for individuals, however, because resources are finite and large groups increase the risk of detection by predators. So there is a trade-off between the advantages and risks inherent to group living. Group size optimization is therefore essential for group

living (Johnson et al., 2002). Our model suggests that multiple interpretations induced by the asynchronous updating method are a natural connection between group formation and decreasing the risk of large groups. In other words, agents in ALM can decrease their risk of food exhausting because each group never stays the same place. Furthermore, we point out the role of weak connection among small groups. Small groups in weak connections can decrease the risk of detection and group's isolation.

### ALMA (ALM with Anticipation)

Now we examine the effect of agent anticipation, defined as incorporating current information with past and future information. The detail of algorithm is listed on **Appendix C**. Current information corresponds to the site pattern of the agent's neighborhood, as discussed concerning the active phase with learning in ALM. To extend this approach, we consider the neighborhood of the neighborhood, in other words all eight neighborhoods of sites reachable in one step. This corresponds to past and future information, because these sites include the site belonged to one step ago, or that which will be belonged to one step from now (see upper left of Figure 4). We can organize these nine site patterns (eight sites for the neighborhood and one for the current agent position). If identical neighborhood patterns exist, we select one of the neighborhoods. These overlapping elements affect the element selection in the next paragraph.

Numbering each neighborhood site (see **Appendix Figure**) enables us to construct a partial order set. Since a partial order set generally never closes with binary operations such as meet and join, this set never constructs a lattice (Davey and Priestelely, 2005). To construct a lattice, we use Dedekind-MacNeille completion (Davey and Priestelely, 2005). Pertinent details from lattice theory are given in **Appendix A and B**. Making a lattice involves constructing a logical relationship, closed under a binary operation, among current, past, and future information.

We then stochastically select an element from the lattice. After selection of an element from the lattice, we next take an ideal from this element and construct a congruence of lattices (Davey and Priestelely, 2005). We applied this method in past research related to topics such as species evolution (Niizato and Gunji, 2013). Roughly speaking, congruence of the lattice means making well-defined groupings on the lattice. This grouping is strongly bound by its lattice structure. We use the group that contains an element of the current information. Since this grouping is also an ordered set, we can certainly pick the top element from this group as a representative element (see upper right of Figure 4). This representative element carries past, current, and future information. Using this method, agents sometimes behave as if they have learned about unlearned situations in the active phase, making the conflict between active and passive phases larger than in ALM.

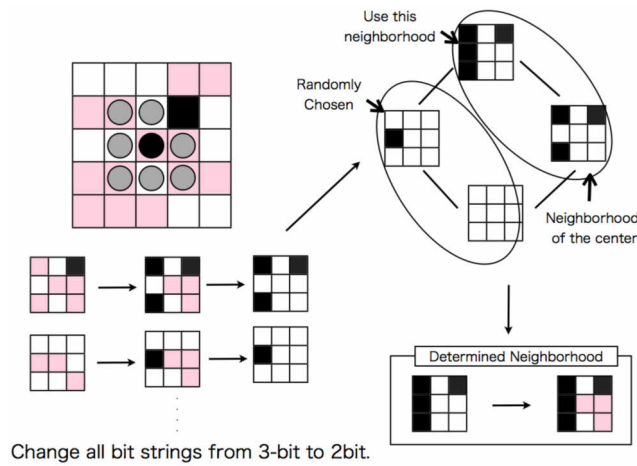


Figure 4: Constructing the lattice. Considering the neighborhood of the neighborhood, we can pick up nine 8-bit strings. The overlapped element will affect the selection of an element, which makes an ideal.

### Sign Making in ALMA

Anticipation of agents contributes to sign making (avoidance sign). We set the environment as in Figure 6 (A). At the center of the boundary, blocks and scent markings are distributed alternately. The first set of scent markings lasts until the end. The region in a given space is divided as Side I and Side II. These blocks prevent free movement between the two sides, but the existence of scent markings attracts agents around the center. In other words, the centerline of this space provides a contradiction for each agent. Do these factors affect the behavior of agents?

To examine the effect of anticipation in this heterogeneous environment, we compare four interaction patterns, namely, the control model, ALM, the control model with anticipation, and ALMA. Figure 5 shows the distribution of the density probability for these four interaction patterns. Obviously, agents of the both of control models (Figure 5 (A) and Figure 5 (C)) concentrate around the blocks, even after many steps have passed. At first glance the behaviors of ALM (Figure 5 (B)) and ALMA (Figure 5 (D)) seem similar, but there is a radical difference between them; in ALMA the density distribution around the centerline is very low compared with ALM and both the control models. This behavior suggests that each agent in ALMA avoids center blocks, and shows territorial behavior in both regions. In other words, each agent in ALMA recognizes a sign of avoidance.

This result is also supported by statistical results (Figure 6(B)). We compared the mean maximum spending time at Side I or Side II with four patterns of interaction (100 times for 100,000 steps). ALMA contains many agents which spend at one side for long time. It is worth to point

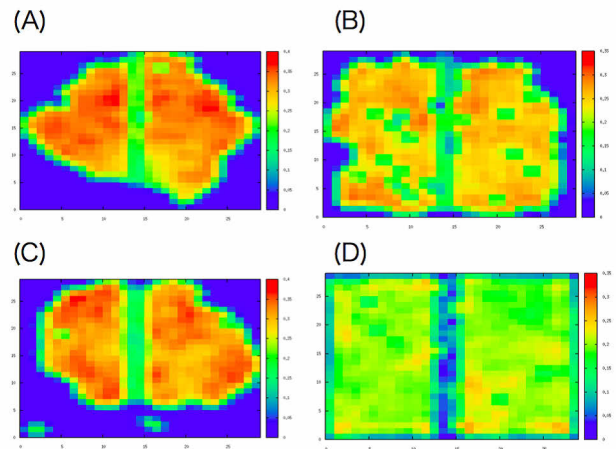


Figure 5: The distribution of density probability for four ways of interactions, the control (upper left), ALM (upper right), the control with anticipation (lower left), and ALMA (lower right). The grid is  $30 \times 30$  and the number of agent is 150.

out that there is almost no difference between the models without anticipation (ALM and the control model). This result suggests that anticipation helps agents recognize signs of avoidance. Furthermore, Figure 6 (B) also suggests that learning in the active phase in ALMA plays an important role for sign making. Because there are many collisions around center blocks, the neighborhood in ALMA contains many occupied sites when the agent is in the active phase. However, this is insufficient for sign making. There are too many neighborhood patterns (6560 patterns!), and since there are conflicts among the past, current, and future information, the constructed lattice would tend to make a modular lattice, meaning that each element has many complementary elements (two elements satisfy  $a \vee b = 1$  and  $a \wedge b = 0$ ). High modularity leads to selection of the top lattice element when we apply the congruence method to the lattice. In this sense, agent's anticipation is driving-force for sign-making in heterogeneous environments.

### Collective Cognition in ALMA

In this section, we discuss the possibility of collective cognition in changing environments. Collective cognition is defined here as a collective response against a changing environment. Cognition, including human cognition, inevitably involves the surrounding situation. We sometimes find differences even between situations that seem to be the same. One famous example is Rubin's vase, a black and white optical illusion. Focusing on one color in the illustration, one sees a vase. Focus on the other, and one sees opposing faces instead. The stimulus from viewing this picture must be the

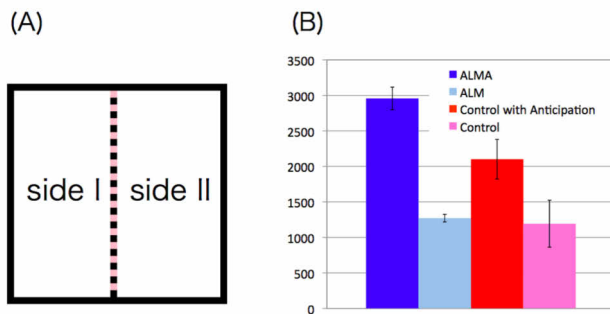


Figure 6: (A) The right side of the center is Side I. The left side of center is Side II. (B) The mean maximum spending time in Side I or Side II. The four bars correspond to the control model (pink), ALM (light blue), the control with anticipation (red), and ALMA (blue).

same, but cognition differs regarding its appearance. Although this is an extreme example, different recognition of the same situation is an essential factor of cognition. Different cognition regarding the same object allows agents multiple interpretations that provoke different reactions against the same object. In this sense, we can discuss an agent's collective cognition in heterogeneous environments.

To examine the possibility of collective cognition in ALMA, we examine the behavior of agents when an environment temporally changes. Here, a changing environment is shown as a changing number of blocks at the centerline, as in the previous section. There are two patterns, additions and subtractions (Figure 7 (A)). Since adding blocks means that blocks suddenly appear in space and removing blocks means gradually removing a wall, these changes are radically different from a cognitive standpoint. We measured the behavioral differences between them as taking the mean density probability around the centerline.

Figure 7 (B) shows the result. The triangles correspond to removed blocks and the rectangles correspond to added blocks. The horizontal line is the rate of blocks in the center, so zero on the horizontal line means there are no blocks in the space. The rate of blocks gradually increases or decreases 0.1 points every 1,000 steps. We performed 100 trials and averaged the value for each block count. Figure 7 (B) suggests that transitions of the density probability are different when adding or removing blocks. The shapes of Figure 7b indicate hysteresis, a property that recalls Harken (1983), a famous study that showed human cognition changes over time.

The density probability around the center gradually decreases with an increased number of blocks. When block rate decreases from 1, however, the density probability drops from 1.0 to 0.5, then increases from 0.5 to 0.0. These differences come from the agent learning in the active phase.

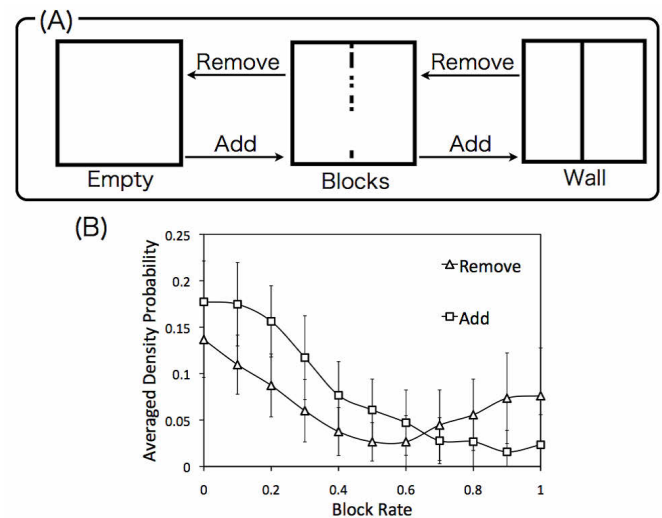


Figure 7: (A) Adding or removing blocks at the center. Three blocks are added or removed every 1,000 steps. (B) The rate of mean density probability around the center with block rates. The triangle corresponds to removing blocks and the rectangle corresponds adding blocks.

Each agent recognizes added blocks as ones that should be interpreted in the active phase. When removing blocks, however, the agent starts to recognize blocks as a part of wall. This wall in itself is not subject to interpretation for each agent, because information about the wall never changes with time. Interpreting blocks as a wall or not a wall changes the agent's behavior.

## Discussion

Self-organization has been discussed by many researchers. To briefly summarize the main assertion of self-organization, it is "simple local interaction that creates global behavior." In real systems, however, this is insufficient to understand emergent phenomena. For example, when we consider the evolution of life, it must have started from simple forms and behaviors. We should therefore ask how simple behaviors evolve into complex ones. A related question would ask about the origin of various higher-layer behaviors. It is considered that the concept of self-organization would answer this question.

However, we have to admit that the concept of self-organization contains a serious problem if we seek the origin of various higher-layer behavior. There are examples even in cellular automata. Class IV automata have computational abilities using patterns of time evolution as particles (Wolfram, 2005). Although Class IV automata can exhibit higher-layer behavior, there is a clear distinction between the system as a device and its modelers. In other words,

the computational property of a cellular automaton never explicitly emerges unless the modelers (or the observer) set the appropriate initial conditions. Even when one high-layer behavior (universal computation in Class IV) emerges, the theory of the self-organization inevitably contains a kind of this problem.

To solve this division problem between the modelers and the systems, we proposed multiplicity of interaction. Introducing multiplicity to simple interaction allows its own interpretation of situation. We notice that multiple interpretations of situations must be distinct from simple hybrid models, which contain many kinds of interactions because the base interaction is consistently single. Having multiple interpretations of a single interaction, the usage of the interaction would be open to many applications without interference from modelers when agents encounter different situations.

In our model, a single interaction corresponds to the attractiveness of scent markers. The multiplicity of interaction was constructed from three factors: asynchronous updating, site pattern learning in the active phase, and anticipation. The latter two concepts originally come from disagreement of timing among agent actions induced by asynchronous updating. Adding the latter two properties, the discrepancy between the active and passive phases becomes larger than the original one. Our model suggests that the discrepancy, induced by asynchronous updating, becomes the driving force of many applications for single interaction in various environments.

We showed that our model could connect various kinds of behaviors, which are sometimes seen as contradictory or irrelevant to each other. First we observed that agents under ALM could search more efficiently than agents under the control model. We discussed that this result had relevance with the ecological searching trade-off problem. Furthermore, agents under ALMA recognize signs of avoidance for the center blocks. Although the behavioral range of agent covers the entire space, each agent avoids the centerline, behaving as if marking territories where they belong. Searching an area means spreading over its entire space. Marking territories means establishing boundaries. These two collective behaviors are qualitatively conflicting, but can be implemented by introducing multiplicity to the interaction without any contradiction. Collective cognition, which we discussed in the last section, is clearly a kind of behavior different from searching behavior. Collective cognition means that agents in ALMA recognize the difference between two ways of changing the environment (adding or removing blocks from the center). Our result suggested that each agent distinguishes between discrete blocks and a wall of blocks.

In this sense, we can conclude that multiplicity of interaction could be open in its usage with various environments. We have confirmed different kinds of global behaviors—searching, sign marking, and collective cognition—by set-

ting various environments using this multiplicity. In other words, to consider how multiplicity of interaction connects the multiplicity of behaviors. If we take this multiplicity of interaction, the degree of the multiplicity would become one possible measure of self-organization of emergent phenomena. Then we can ask the origin of complex behaviors on living systems.

## References

- Brooks, R. (1990). Elephants don't play chess. *Robotics and Autonomous Systems*, 6:3–15.
- Cornforth, D., Green, D. G., and Newth, D. (2005). Ordered asynchronous processes in multi-agent systems. *Physica D: Nonlinear Phenomena*, 204:70–82.
- Couzin, I. D. (2009). Collective cognition in animal groups. *Trends in cognitive sciences*, 13(1):36–43.
- Davey, B. and Priestelely, H. (2005). *Introduction to Lattices and Order*. Cambridge University Press, Cambridge.
- Giuggioli, L., Potts, J. R., and Harris, S. (2011). Animal interactions and the emergence of territoriality. *PLoS computational biology*, 7:e1002008.
- Gunji, Y.-P., Haruna, T., and Sawa, K. (2006). Principles of biological organization: Local-global negotiation based on “material cause”. *Physica D: Nonlinear Phenomena*, 219:152–167.
- Haken, H. (1983). *Synergetics, an Introduction: Nonequilibrium Phase Transition and Self-Organization in Physics, Chemistry and Biology*. New York: Springer-Verlag.
- Jackson, A. L. (2006). Toward an individual-level understanding of vigilance: the role of social information. *Behavioral Ecology*, 17:532–538.
- Johnson, D. D., Kays, R., Blackwell, P. G., and Macdonald, D. W. (2002). Does the resource dispersion hypothesis explain group living? *Trends in Ecology & Evolution*, 17:563–570.
- Lewis, M. and Murray, J. (1993). Modelling territoriality and wolf-deer interactions. *Nature*, 366:738–740.
- Niizato, T. and Gunji, Y.-G. (2013). Interactions between species and environments from incomplete information. *Biosystems*, 111:145–155.
- Parrish, J. K. (1999). Complexity, Pattern, and Evolutionary Trade-Offs in Animal Aggregation. *Science*, 284:99–101.
- Sumpter, D. J. T. (2006). The principles of collective animal behaviour. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 361:5–22.
- Wolfram, S. (2005). *New Kind of Science*. Wolfram Media Inc.


## Appendix A

We briefly introduce the lattice theory for unfamiliar readers of the lattice theory.

**Definition 1 (Partial Order)** Let  $P$  be a set. An order on  $P$  is a binary relation  $\leq$  on  $P$  such that, for all  $x, y, z \in P$ .

- (i)  $x \leq x$



1	2	3
4		5
6	7	8

**Appendix Figure** : The way of numbering the neighborhood.

- (ii)  $x \leq y$  and  $y \leq x$ , then  $x = y$
- (iii)  $x \leq y$  and  $x \leq z$ , then  $x \leq z$

We denote a partially ordered set by the pair,  $(P, \leq)$ . For example, a set of bit (binary) strings can construct a partial order. A bit string  $a_1 a_2 \dots a_n$  is a finite sequence of zero, one ( $a_i \in \{0, 1\}$ ). An order between two bit strings such as  $a_1 a_2 \dots a_n$  and  $b_1 b_2 \dots b_n$  is defined by  $a_1 a_2 \dots a_n \leq b_1 b_2 \dots b_n$  if  $a_i \leq b_i$  for all  $i$ . We use a set of bit strings in this study. However, A partial order is not a lattice. Then we define the meet and the join. We define the join " $\vee$ " and the meet " $\wedge$ " of two elements  $x$  and  $y$  in  $P$ . The join can be defined by  $x \vee y = \sup\{x, y\}$  when it exists. The join can be defined by  $x \wedge y = \inf\{x, y\}$  when it exists. The notation of  $\sup(\inf)$  means the lowest (greatest) upper bound of  $\{x, y\}$  in  $P$ .

**Definition 2 (Lattice)** Let  $(P, \leq)$  be a non-empty partially ordered set. If  $x \vee y$  and  $x \wedge y$  exist for all  $x, y \in P$ , then  $(P, \leq)$  is called for a lattice.

**Definition 3 (Ideal)** Let  $(L, \leq, \wedge, \vee)$  be a lattice. A non-empty subset of  $J$  is called an ideal if

- (i)  $x, y \in J$  implies  $x \vee y \in J$ ,
- (ii)  $x \in L, y \in J$  and  $x \leq y$  imply  $x \in J$ .

**Definition 4 (Congruence on a Lattice)** Let  $(L, \leq, \wedge, \vee)$  be a lattice. Let an equivalence relation on  $L$  be  $\theta = \{< x, y > \in L \times L \mid \text{such that any } x, y, z \in L,$

- (i)  $< x, x > \in \theta$
- (ii)  $< x, y > \in \theta \Leftrightarrow < y, x > \in \theta$
- (iii)  $< x, y > \in \theta$  and  $< y, z > \in \theta \Rightarrow < x, z > \in \theta$

We also denote  $< x, y > \in \theta$  as  $xy \pmod{\theta}$ . Then an equivalence relation is a congruence on  $L$ , if for any  $x, y, z, w \in L$ , ( $x \equiv y \pmod{\theta}$  and  $z \equiv w \pmod{\theta}$ )  $\Rightarrow$  ( $x \vee z \equiv y \vee w \pmod{\theta}$  and  $x \wedge z \equiv y \wedge w \pmod{\theta}$ ).

More detail in Davey and Priestelely in 2005.

**Theorem (Reconstruction of a Lattice from a Quotient Lattice)** Let  $L$  be a lattice and  $f$  be a natural quotient map such as  $f : L \rightarrow L/\theta$ . For the binary relation derived from an ideal  $J \subseteq L$ , there exists a filter  $K \subseteq L$  such that  $[x]_{\theta(J)} = f^{-1}(x)$ , where for any  $x \in K$ ,  $f^{-1}(x) := \downarrow x - \cup_{y \in K, y < x} \downarrow y$

**Proof** See in (Gunji et al., 2006; Niizato and Gunji, 2013).

## Appendix B

To construct the lattice from a partial order set, we use Dedekind-MacNeille completion, as follows: In the partial order set  $P$ , we can take a lower and upper closed set, which we represent as  $P^u$

(upper) and  $P^l$  (lower). The upper set is  $P^u = \{x \mid y \leq x \text{ for any } y \in S\}$ . The lower set  $P^l$  is the dual of  $P^u$ .

**Theorem (Dedekind-MacNeille completion)** For any order set  $P$ , we can construct the set as follows:

$$\text{DM}(P) = \{A \subseteq P \mid A^{ul} = A\}$$

Generally, a set that satisfies  $A^{ul} = A$ , is called a "cut". To understand the meaning of Dedekind-MacNeille completion, the following lemma is useful.

**Lemma** For any ordered set  $P$ , the following statements are satisfied.

- (i) For any  $x \in P$ ,  $(\downarrow x)^{ul} = \downarrow x$ .
- (ii) For any  $A \subseteq P$  and  $\bigvee A \in P$ , then  $A^{ul} = \downarrow (\bigvee A)$ .

## Appendix C

Here we describe the detail of the algorithm of ALM and ALMA. This model consists of  $N$  agents moving in discrete time and in a grid space. Numbering neighbor sites as **Appendix Figure**, we can define 3-bit strings as  $\bar{a} = a_1 a_2, \dots, a_n$ . We also define the neighborhood of the neighborhood in ALMA as eight 3-bit strings  $\bar{b}_1, \bar{b}_2, \dots, \bar{b}_8$ . First, agents are randomly distributed in a given space. The algorithm is constructed as follows.  $i$  is a tag of the agent. Each agent is chosen asynchronously in density dependent.

**For** ( $\exists j \in \{1, 2, \dots, 8\} \mid a_j^i$  is occupied) {  
 $\text{Memo}^i(a^i) = a^i$

Select the moving site from a set  $S$ .

$$S := \{j \mid a_j^i = 0 \text{ or } a_j^i = 1\}$$

The selection probabilities are;

$$\text{Prob}(a_j^i = 0) = \mu / \sum_j \text{Weight}(a_j^i)$$

$$\text{Prob}(a_j^i = 1) = 1 / \sum_j \text{Weight}(a_j^i)$$

}  
**For** ( $\forall j \in \{1, 2, \dots, 8\} \mid a_j^i$  is not occupied) {

**If** (Using ALM) {

$$\bar{a}^i \leftarrow \text{Memo}^i(\bar{a}^i)$$

Select the moving site from a set  $S$ .

$$S := \{j \mid a_j^i = 0 \text{ or } a_j^i = 1\}$$

The selection probabilities are;

$$\text{Prob}(a_j^i = 0) = \mu / \sum_j \text{Weight}(a_j^i)$$

$$\text{Prob}(a_j^i = 1) = 1 / \sum_j \text{Weight}(a_j^i)$$

}

**If** (Using ALMA) {

$$\bar{b}_0^i \leftarrow \text{Memo}^i(\bar{a}^i)$$

Pick up the neighborhood of neighborhood of  $i$ ;

$\bar{b}_1^i, \bar{b}_2^i, \dots, \bar{b}_8^i$  respectively.

**For** ( $0 \leq j \leq 8$ ) {

**For** ( $1 \leq k \leq 8$ ) {

$$\text{If} (\bar{b}_{j,k}^i = 1) \bar{b}_{j,k}^i = 0$$

$$\text{Else If} (\bar{b}_{j,k}^i = 2) \bar{b}_{j,k}^i = 1$$

}

}

Make a partial order set  $P$  from  $\{\bar{b}_0^i, \bar{b}_1^i, \dots, \bar{b}_8^i\}$ .

$$P := \{s_1, s_2, \dots, s_m\}$$

$m (\leq 9)$  is the number of elements of  $P$ .

Count the overlapping elements.

$$Num(s_1) = n_1, \dots, Num(s_m) = n_m$$

Make a lattice  $L$  from  $P$

Using **DM** completion.

$$L = \{s_1, s_2, \dots, s_m, s_{m+1}, \dots, s_{m+l}\}$$

$l$  is the number of adding elements.

Select one element from  $L$ .

The selection probability of each element is,

$$Prob(s_k) = Num(s_k) / \sum_j Num(s_j)$$

Make a congruence by using selected element  $s_k$ .

Then select the top element in the congruence.

Mathematically, when  $J$  is an ideal,

$$s''^i \Leftarrow \bigvee \{s_j \in L | s_j \in [s_0]_{\theta(J)}\}$$

**For** ( $1 \leq j \leq 8$ ) {

**If** ( $s_j'' = 1$  and  $b_j^0 = 1$ )  $a_j^i = 2$

**Else If** ( $s_j'' = 0$  and  $b_j^0 = 0$ )  $a_j^i = 0$

**Else**  $a_j^i = 1$

}

Select the moving site from a set  $S$ .

$$S := \{j | a_j^i = 0 \text{ or } a_j^i = 1\}$$

The selection probabilities are;

$$Prob(a_j^i = 0) = \mu / \sum_j Weight(a_j^i)$$

$$Prob(a_j^i = 1) = 1 / \sum_j Weight(a_j^i)$$

}

$Memo^i(-)$  means the store information about site pattern for each agent  $i$ .  $Num(-)$  means the number of overlapping element in  $\{b_0^i, b_1^i, \dots, b_8^i\}$ .

# Task Allocation Strategy for Time-Constrained Tasks in Robots Swarms

Yara Khaluf<sup>1</sup> and Franz Rammig<sup>1</sup>

<sup>1</sup>Heinz Nixdorf Institut, University of Paderborn, Germany  
{yara, franz}@hni.uni-paderborn.de

## Abstract

Task allocation is a key problem, which has a direct influence on the system performance in all kinds of distributed systems. This paper focuses on a specific kind of task allocation in swarm robotic systems, where the tasks are associated with specific time constraints.

The paper presents a self-organized task allocation strategy, which aims to assign robot swarms to time-constrained tasks in a distributed manner. The robots assignment is performed based on particular specifications including task sizes and deadlines in addition to the specification of the single robot performance on the considered tasks. No central control is required to govern the swarm behaviour and no communication is exploited among robots.

## 1 Introduction

Swarm robotics is a recent field of research that takes inspiration from complex natural systems such as colonies of social insects (ants, honeybees, etc.) or groups of cooperating animals. It is a kind of mobile distributed system with a high density and which can be mainly characterized by its: redundancy, where robots failures do not affect the system functionality, scalability, since the system can be extended by adding robots, and flexibility where it can be used to perform a large spectrum of applications.

In many practical robotics applications the successful execution of a task depends not only on the logical correctness of the operations that robots are performing, but also on the time before which the results are delivered. Such tasks are referred to as real-time tasks and they are generally categorized according to their deadlines into: *hard* real-time tasks, where missing the deadline can lead to catastrophic results and *soft* real-time tasks, where missing the deadline decreases the quality of results. Real-time tasks will be common to encounter as soon as the swarm robotic systems are exported out of the research labs to be involved in real life applications. While dealing with hard-deadlines is beyond the capabilities of a fully stochastic system like swarm robotics, tasks with soft deadlines are the suitable candidates for swarm robotics. This paper discusses real-time tasks with soft deadlines, which need to be performed

by a swarm of homogeneous robots. The goal is to assign the robots to the tasks under their time constraints in a fully distributed and autonomous manner. The proposed allocation strategy neglects initially the influence of physical interferences among robots, on the overall performance of the system. However the strategy can be extended later to include this influence.

The rest of the paper is organized as follows: section 2 reviews the literature of task allocation in swarm robotic systems with and without time constraints. In section 3 a formulation of the task allocation problem under time constraints is introduced. The designed allocation strategy is explained with its different stages in section 4. Section 5 presents a numerical example with its Monte-Carlo simulations to illustrate the steps of the allocation strategy and verify it. In section 6 a swarm robotic scenario is introduced where the allocation strategy is applied and simulated. The paper is concluded in section 7.

## 2 Related Work

Task allocation can be found in nature such as in ant and bee colonies Bonabeau et al. (1998). Mathematical models of task allocation, which focus on simple reactive mechanisms and study the fraction of robots engaged in a particular task as a function of the number of available tasks as perceived by the robots, were performed like Lerman et al. (2006). The task allocation solutions proposed in the literature can be classified in three broad categories: *centralized*, *negotiation-based* and *self-organized*: centralized techniques assume the presence of a central coordinator responsible for the allocation of the agents to the tasks. Self-organized systems, on the contrary, are constituted by peers that take decisions autonomously, with limited negotiations with other peers and without a central point of control. This kind of systems are generally less prone to catastrophic failures and considered a better approach when rapid adaptation to changes in the environment is required. Most of these studies tackle simple problems without task interdependencies, Dahl et al. (2009). Negotiation-based approaches, generally based on auction-based strategies, are the compromise solution be-

tween centralized and self-organized systems, Dias et al. (2005), Gerkey and Mataric (2002), Zheng et al. (2006). In auction-based strategies, the robots bid on the announced task according to specific task characteristics and to their relative capabilities. One of the characteristics that is often criticized in this approach, is that many negotiation-based solutions assume a fully connected network among the robots, which is not the case in many realistic applications. A comparison between the auction-based approaches and the self-organized ones based on threshold can be found in Kalra and Martinoli (2006). A general taxonomy of task allocation strategies in robotic systems has been presented in Gerkey and Mataric (2004) along three main comparisons: *single-task robots (ST)* vs. *multi-task robots (MT)*, where single and multiple robots are able to perform only one task at time; *single-robot tasks (SR)* vs. *multi-robot tasks (MR)*, where each task needs one robot or more, and the *instantaneous assignment (IA)* vs. *time-extended assignment (TA)* where it is assumed that the robots and the environment allow only for an instantaneous task allocation with no future planning.

In swarm robotics, response-threshold mechanisms are relatively common Nouyan et al. (2005), Nouyan et al. (2004), Ducatelle et al. (2009b), Ducatelle et al. (2009a), Krieger and Billeter (2000). In this approach, each robot is programmed to react to stimuli associated to the different tasks. Agassounon and Martinoli (2002) introduce a task allocation for the traditional swarm task "foraging". Another threshold-based algorithm for allocating workers to a given task whose demand evolves dynamically over time is presented in Agassounon et al. (2001). In Liu et al. (2007) a mathematical model for a similar task allocation behaviour is introduced. Some works combine the common swarm response-threshold approach with a kind of communication protocol to avoid the need for a central unit as in Zhang et al. (2007). A very few works to our best knowledge, have assumed a target distribution for the robots over all the available tasks to be reached like we can find in McLurkin and Yamins (2005). Few authors have studied the problem of task allocation in swarm robotic systems with time constraints associated to tasks. Some of the performed studies were based on the auction techniques for the allocation in respect to deadlines like in Guerrero and Oliver (2010) and Guerrero and Oliver (2011). Other works like Acebo and Rosa (2008), have introduced a heuristic based on the so-called Bar-System model, where the key idea is to simulate the way waitresses assign themselves to bar customers in an efficient and distributed way. The approach is then applied to a group of loading robots for a commercial harbour. In Schneider et al. (2005) and Jones et al. (2007) market-based task allocation strategies, where time is the critical constraint, are considered together with a reward mechanism associated to a task being successfully completed.

### 3 Problem Formulation

The problem of autonomous task allocation in presence of deadlines can be formulated as follows: a swarm of  $N$  robots should be allocated to a set of  $m$  tasks  $\{T_1, \dots, T_m\}$ . The task deadlines  $\{D_1, \dots, D_m\}$  and the task sizes  $\{S_1, \dots, S_m\}$ , are assumed to be known a priori. The task is assumed to be built up of individual parts, where the robot can accomplish one part per time. The size of any task represents the discrete number of parts which should be accomplished within the task deadline. Each task  $T_i$  is composed of  $S_i$  parts and accomplishing  $T_i$  is achieved by accomplishing all of its  $S_i$  parts. The real-time tasks we consider in our study have soft deadline and require to be executed in parallel. The switching costs between them are negligible in comparison to the task deadlines. The system is designed as a fully autonomous one, where no communication among robots is applied and no central unit is used for the allocation purposes.

The single robot performance on a specific task is expressed in terms of the random time required by the robot to accomplish one part of the considered task. The random variable associated with the single robot performance is modelled in this paper as a normal distributed variable with a task-specific mean and a task-specific standard deviation. The single robot performance is an essential input for the developed allocation strategy. Robots can measure their individual performances by working on each of the considered tasks for a specific period of time, registering the times they require to accomplish individual parts and estimate the mean and standard deviation related to their performance on each task. Tasks are served according to their priorities, which are derived based on the task deadlines. The task with a shorter deadline has a higher priority to be executed. Before starting the execution, a list of the  $m$  tasks with their sizes and deadlines is provided to the swarm. These task specifications in addition to the single robot performance on the different tasks are the inputs used later to perform the tasks allocation.

We concentrate in this paper on a kind of *dynamic* task allocation, where the robot is allowed to stay on the same task or to switch to another one. The switching decision could be taken each time the robot finishes working on a part of the current task or at specific time points. Switching at specific time points, during the execution times of the tasks, requires global synchronization among the robots to take the decision at the specified time point. Dynamic task allocation is useful to be applied in many applications, where the switching costs among tasks can be considered being negligible. Such cases can be encountered when the tasks occupy a shared physical arena, so robots do not need to travel among them while switching. An example is foraging multiple kinds of objects where each kind represents an individual task and all kinds are scattered on the same arena. Another possibility to omit switching costs, is when they are negligi-



ble in comparison to the task deadlines.

The selection of the next task is performed by means of allocation probability matrices, which are calculated based on the task specifications and the single robot performance. The probability matrices are of the form  $P_{ij}$ , which represents the probability to switch to task  $T_j$  from task  $T_i$  while the switching costs are negligible in comparison to the deadlines.

#### 4 Robot Allocation Strategy

The goal of this study is to develop a feasible task allocation strategy that allows robots to assign themselves autonomously to a set of real-time tasks with respect to the task deadlines. In a fully stochastic system like swarm robotics where even the performance of a single robot on a specific task represents a stochastic variable, it is particularly difficult to develop an allocation strategy which can guarantee the execution of the task within its deadline.

Our strategy attempts to find an optimal number of robots to be assigned to each task based to the size and deadline of the task in addition to the performance of the single robot on that task. The required number of robots on each task is used to derive allocation probability matrices, which are used by the robots to allocate themselves autonomously to the tasks during their execution.

Each task is considered to be "inactive" as soon as it is completely accomplished or when its deadline is exceeded, otherwise the task is considered as "active". Robots are not required to be assigned to inactive tasks. Consequently, the number of robots available for allocation changes based on the current number of active tasks. In order for the allocation strategy to exploit the current number of robots available for allocation, the time between the start of tasks execution and the largest deadline, is divided into periods:  $\{\pi_1, \dots, \pi_m\}$  where:

$$\pi_i = D_i - D_{i-1} \quad \forall i \in \{2, \dots, m\} \quad (1)$$

The first period has the length of the earliest deadline  $\pi_1 = D_1$ . Figure 1 illustrates the periods and the active tasks within each period.

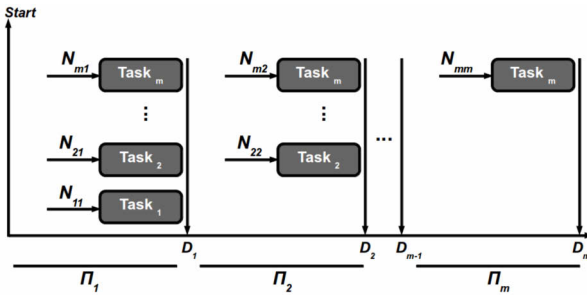


Figure 1: Active tasks over the defined periods

#### 4.1 Required Number of Robots

The single robot performance on task  $T_i$  is expressed, as mentioned above, in terms of the time required by a single robot to accomplish one part of task  $T_i$ . This random time is modelled as a normally distributed variable with the mean  $\mu_i$  and the standard deviation  $\sigma_i$ . Let us use  $k_i$  to denote the number of parts which could be accomplished by a single robot on task  $T_i$  within its deadline  $D_i$ . The value of  $k_i$  is taken within the discrete range  $[0, +\infty[$ , which represents the possible outcomes related to the number of parts could be accomplished by a single robot within  $D_i$ . Let us define the event  $E_i(k_i)$  that a single robot accomplishes  $k_i$  parts on task  $T_i$  within  $D_i$ . We refer to the time spent by a single robot to accomplish  $k_i$  parts by  $\tau_i(k_i)$ , then we have the following two events equivalents:

$$E_i(k_i) \iff \tau_i(k_i) \leq D_i \quad (2)$$

The probabilities of the equivalent events in Eq. (2) are equal and they represent the probabilities we are interested in:

$$\Pr(E_i(k_i)) = \Pr(\tau_i(k_i) \leq D_i) \quad (3)$$

As the time spent by the robot to accomplish *one* part of task  $T_i$  is normally distributed with the mean  $\mu_i$  and the standard deviation  $\sigma_i$ , the right side of Eq. (3) is the probability that the sum of  $k_i$  random variable, each one being distributed using  $Norm(\mu_i, \sigma_i)$ , is smaller than or equal to  $D_i$ . It is well known that the sum of  $n$  random variable each one being distributed with  $Norm(\mu, \sigma)$ , is a random variable distributed normally with the mean  $n\mu$  and the standard deviation  $\sqrt{n}\sigma$ . Consequently, the probability  $\Pr(\tau_i(k_i) \leq D_i)$  in Eq. (3) represents the cumulative density function CDF of the normal distribution with the mean  $k_i\mu_i$  and the standard deviation  $\sqrt{k_i}\sigma_i$ .

$$\Pr(\tau_i(k_i) \leq D_i) = \frac{1}{2} [1 + \operatorname{erf}(\frac{D_i - k_i\mu_i}{\sqrt{2k_i}\sigma_i})] \quad (4)$$

The allocation strategy applies the cumulative density function in Eq. (4) to find out the probability associated with the event  $E_i(k_i)$  for each  $k_i \in [0, S_i]$ , where  $S_i$  is the size required to be accomplished on  $T_i$  within  $D_i$ . This probability  $P(E_i(k_i))$  is referred to as the success probability of the event  $E_i(k_i)$ . The events  $E_i(k_i)$  are distributed following a binomial distribution with the success probabilities  $P(E_i(k_i))$ , see Figure 2.

The expected value of a random variable  $X$  which is distributed according to a binomial distribution with  $n$  trials and the success probability  $p$  is given by:

$$E[X] = np \quad (5)$$

We map the number of trials  $n$  to the required number of robots, where each robot can accomplish  $k_i$  parts with a success probability  $P(E_i(k_i))$ . In order to find the required size

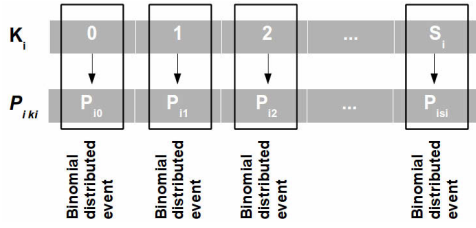


Figure 2:  $E_i(k_i)$  for each  $k_i \in [0, S_i]$  with the success probability associated to each of  $k_i$  values

of trials (number of robots), such that one robot of the  $n$  robots is expected to accomplished  $k_i$  parts within  $D_i$ , we substitute the expected value by 1 and the success probability by  $P(E_i(k_i))$  in Eq. (5), so we have:

$$1 = n_i P(E_i(k_i)) \implies n_i = \frac{1}{P(E_i(k_i))}$$

$n_i$  is the number of robots required to achieve an expected number of accomplished parts equal to  $k_i$  within  $D_i$  by one robot. However, we aim to achieve an expected number of accomplished parts equal to  $S_i$  within  $D_i$ , hence the required number of robots is calculated using the following equation:

$$N_i = \lceil \frac{n_i S_i}{k_i} \rceil \quad (6)$$

Let us consider the example of one task with the size  $S_i = 10$  parts, the deadline  $D_i = 10$  time units. The single robot performance on this task in terms of the time required by the single robot to accomplish one part, is normally distributed. We assume different means of the random time:  $\mu_i \in \{2, 4, 6\}$  and a unique standard deviation  $\sigma_i = 0.01$ . The number of parts  $k_i$ , which could be accomplished by a single robot within  $D_i$  can takes its value in  $[0, 1, \dots, +\infty]$ . However the range of interest is  $k_i \in \{0, 1, \dots, 10\}$ . Each event  $E_i(k_i)$  of accomplishing  $k_i$  parts by a single robot is associated with a probability calculated using Equation (4). Figure 3 shows the different values of  $k_i$  with their success probabilities calculated for the different means. We consider the mean  $\mu_i = 2$  for the rest of the example. Figure 4 shows how the required size of trials  $n_i$  changes with changing  $k_i$ .  $n_i$ , represents the size of the robots needed to have on average one robot accomplishing  $k_i$  parts within  $D_i$ . In the same figure we can see the total number of robots  $N_i$  required by task  $T_i$  calculated for each  $k_i$  value.

#### 4.2 Optimal Robot Number and Allocation Probability Matrices

In the previous section the allocation strategy calculates the required number of robots  $N_i$ , which should be assigned to task  $T_i$  to accomplish  $S_i$  parts on average within the deadline  $D_i$ . There exist several possible number of robots  $N_i$  to be used. Each value of the  $N_i$  is associated with a unique value

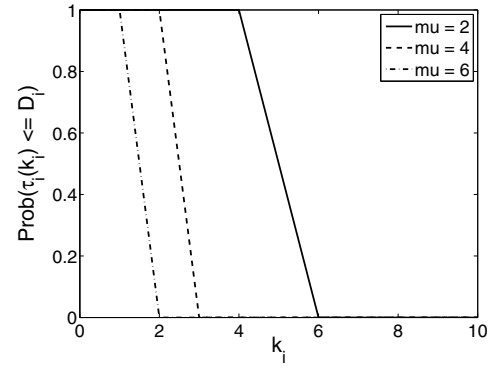


Figure 3: The relation between the number of parts could be accomplished by a single robot within the deadline and its probability of success

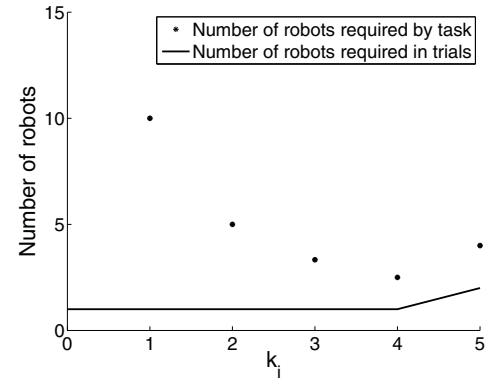


Figure 4: The relation between the number of parts  $k_i$  and the number of required robots in trials  $n_i$  and by the task  $N_i$

of parts,  $k_i$ . The process starts by finding the success probability related to each event  $E_i(k_i)$ , where  $k_i \in \{0, \dots, S_i\}$ . After that, the size of the trials (robots), which is required to have on average one robot accomplishing  $k_i$  parts within  $D_i$ , is determined. Finally the total number of required robots is calculated using Equation (6).

However, a robot swarm represents a limited resource with a given size, which applies a strict constraint on selecting the feasible number of robots  $N_i$ , could be assigned to each of the active tasks during period  $\pi_j$ :

$$\sum_{i=j}^m N_i \leq N \quad (7)$$

where  $N$  is the size of swarm used in the solution.

The developed strategy attempts to minimize the number of robots required by each task individually. Assigning the minimum of the feasible numbers of robots, reduces the impact of potential physical interactions and save robotic resources from unnecessary use. In addition, it provides a

higher chance to schedule newly arrived tasks. Hence, the allocation strategy aims to minimize, for all tasks, the objective function introduced in Eq. (6) under the constraint of Eq. (7) and consequently to find the optimal number of robots for any task  $T_i$ :

$$N_i^{opt} = \min(\lceil \frac{n_i}{k_i} S_i \rceil) \quad (8)$$

The optimal number of robots is the robots number, which the allocation strategy aims to assign to each task though all the periods where this task is active. However, as the swarm size is limited to  $N$  robots, it is possible to not have enough robots in order to assign  $N_i^{opt}$  to task  $T_i$  over all its active periods. The allocation strategy starts, for each period, to satisfy the robot needs of the tasks according to their priorities, which are based on their deadlines. Lack of robots can occur on task  $T_i$  at any of its periods and this robot lack leads, in turn, to a lack in the amount of work was planned to be accomplished by the  $N_i^{opt}$  robots on task  $T_i$  within the considered period. Let us denote the missed number of robots on task  $T_i$  within the period  $\pi_j$  by  $\delta_i(\pi_j)$ . This number of robots was missed to work on task  $T_i$  within the time  $\tau(\pi_j)$ , which is the length of period  $\pi_j$ . We introduce the term of *robots lack density* to refer to the lack of robots happened over a specific period of time like  $\tau(\pi_j)$ . Let us denote the lack density happened on task  $T_i$  during period  $\pi_j$ , by  $\alpha_i(\pi_j)$ . This lack density can be calculated as in following:

$$\alpha_i(\pi_j) = \delta_i(\pi_j) \tau(\pi_j) \quad (9)$$

The lack density associated with all robot lacks, which happened on task  $T_i$  up to period  $\pi_j$  is denoted by  $\alpha_i^{\pi_j}$  and represents the sum of the lack densities over all the task periods up to  $\pi_j$ :

$$\alpha_i^{\pi_j} = \sum_{k=1}^{j-1} \delta_i(\pi_k) \tau(\pi_k) \quad (10)$$

The goal now, while finding out the required number of robots to be assigned to task  $T_i$  during period  $\pi_j$ , is to cover the robot lacks happened on task  $T_i$  up to period  $\pi_j$ . The needed number of robots to cover the lacks that happened on  $T_i$  up to period  $\pi_j$ , should be scaled based on the length of  $\pi_j$ . The sum of lack densities associated with the previous periods is used to find out the number of robots,  $\delta_i^{\pi_j}$ , required to cover the lacks of the previous periods:

$$\delta_i^{\pi_j} = \lceil \frac{\alpha_i^{\pi_j}}{\tau(\pi_j)} \rceil \quad (11)$$

Finally, the allocation strategy can calculate the number of robots required to be assigned to task  $T_i$  during period  $\pi_j$  like follows:

$$N_i(\pi_j) = \begin{cases} N_i^{opt} + \delta_i^{\pi_j} & \text{if } N_{current} \geq N_i^{opt} + \delta_i^{\pi_j} \\ N_{current} & \text{if } N_{current} < N_i^{opt} + \delta_i^{\pi_j} \end{cases} \quad (12)$$

After calculating the number of robots to assign to each task over the periods where the task is active, the allocation strategy outputs a set of probability matrices associated with the defined periods. These matrices are used for the dynamic allocation of the robots. Robots use the probability matrix of the current period to allocate themselves to the tasks: each time a robot finishes working on one part of the current task, or at the beginning of current period all robots allocate themselves to the active tasks using the matrix of the period. This dynamic allocation is a self-organized process, as no central unit controls the robots for assignment. The allocation probability  $P_i(\pi_j)$  of task  $T_i$  in period  $\pi_j$  will be then calculated using the following equation:

$$P_i(\pi_j) = \frac{N_i(\pi_j)}{\sum_{k=j}^m N_i(\pi_k)} \quad (13)$$

where  $P_i(\pi_j)$  is the probability to switch from any task to task  $T_i$  in period  $\pi_j$ .

## 5 Numerical Example

We introduce in this section a numerical example to illustrate the mechanism of the developed allocation strategy. Let us assume to have a set of 5 tasks  $\{T_1, T_2, T_3, T_4, T_5\}$  with the soft deadlines  $\{30, 90, 150, 250, 500\}$  and the sizes  $\{1000, 2000, 3000, 4000, 5000\}$ . A homogeneous swarm of  $N = 400$  is used to execute the tasks, where the performance of any individual robot belonging to this swarm on each task is normally distributed with a task-specific mean  $\mu_i$  and a task-specific standard deviation  $\sigma_i$ . For our example the means of the single robots performance on the 5 tasks are as following  $\{2, 3, 4, 2, 6\}$  and the standard deviations are  $\{0.2, 0.3, 0.1, 0.01, 0.5\}$ .

The possible outcomes in terms of task parts, which could be accomplished by a single robot within the deadline on the different tasks, are:  $k_1 \in \{0, 1, \dots, 1000\}$ ,  $k_2 \in \{0, 1, \dots, 2000\}$ ,  $k_3 \in \{0, 1, \dots, 3000\}$ ,  $k_4 \in \{0, 1, \dots, 4000\}$ ,  $k_5 \in \{0, 1, \dots, 5000\}$ . Each event  $E_{k_i}$  of accomplishing  $k_i$  parts by a single robot on task  $T_i$  is associated with a success probability, that is calculated using Equation (4).

The allocation strategy selects the minimum number of robots to be assigned to each of the considered tasks during the periods where the tasks are active:

$$T_1 = 72 \quad T_2 = 72 \quad T_3 = 82 \quad T_4 = 33 \quad T_5 = 62$$

The sum of the required robots numbers is verified against the constraint in (7):

$$72 + 72 + 82 + 33 + 62 \leq 400$$

The lack of robots may occur on any of the 5 tasks is 0 over all periods, as the swarm size is large enough to cover their

robot needs. Hence, the required number of robots to be assigned to task  $T_i$  on any of its periods is given by:

$$N_i(\pi_j) = N_i^{opt}$$

The probability to apply by each robot for assigning itself to task  $T_i$  during period  $\pi_j$  is given by:

$$P_i(\pi_j) = \frac{N_i(\pi_j)}{\sum_{k=j}^m N_i(\pi_k)} = \frac{N_i^{opt}}{\sum_{k=j}^m N_k^{opt}}$$

Thus, the probability matrices over the 5 periods are as in following:

Period  $\pi_1$ :

$$\begin{pmatrix} P_1(\pi_1) & P_2(\pi_1) & P_3(\pi_1) & P_4(\pi_1) & P_5(\pi_1) \\ 72/321 & 72/321 & 82/321 & 33/321 & 62/321 \end{pmatrix}$$

Period  $\pi_2$ :

$$\begin{pmatrix} P_1(\pi_2) & P_2(\pi_2) & P_3(\pi_2) & P_4(\pi_2) & P_5(\pi_2) \\ 0 & 72/249 & 82/249 & 33/249 & 62/249 \end{pmatrix}$$

Period  $\pi_3$ :

$$\begin{pmatrix} P_1(\pi_3) & P_2(\pi_3) & P_3(\pi_3) & P_4(\pi_3) & P_5(\pi_3) \\ 0 & 0 & 82/177 & 33/177 & 62/177 \end{pmatrix}$$

Period  $\pi_4$ :

$$\begin{pmatrix} P_1(\pi_4) & P_2(\pi_4) & P_3(\pi_4) & P_4(\pi_4) & P_5(\pi_4) \\ 0 & 0 & 0 & 33/95 & 62/95 \end{pmatrix}$$

Period  $\pi_5$ :

$$\begin{pmatrix} P_1(\pi_5) & P_2(\pi_5) & P_3(\pi_5) & P_4(\pi_5) & P_5(\pi_5) \\ 0 & 0 & 0 & 0 & 62/62 \end{pmatrix}$$

We simulate the behaviours of the robots using a Monte-Carlo simulation which was repeated 500 times for the tasks specified in the example above. We assume a global synchronization among the robots. Thus, the allocation probability matrices calculated above, are used by robots each at the beginning of its related period to allocate themselves to the tasks during the whole period. Figure 5 shows a comparison between the average of the total number of parts accomplished by the swarm on each of the 5 tasks and the task size.

## 6 Robotic Scenario

In this section we introduce a multi-task scenario where a homogeneous swarm of simple robots is used to work on a multi line production system. In the considered system, different kinds of objects, which are located initially in objects repositories, Figure 6, should be transported to their production areas. We assume 3 arenas of different sizes, where the robot swarm is used to accomplish the transportation tasks on the different arenas. Each arena is associated with a task of transporting a specific number of objects from their repositories to their production area within a specific deadline. The task sizes are:  $\{70, 90, 110\}$  objects and the task deadlines are:  $\{30000, 50000, 90000\}$  time units, where each time unit in our simulation represents 1/10 of a second. The total size of the used swarm is  $N = 20$

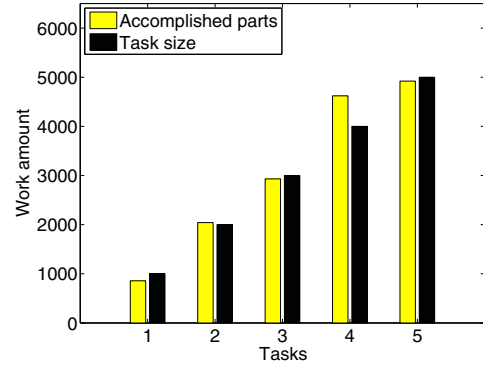


Figure 5: A comparison between the average of accomplished parts resulted from 500 repeat of Monte-Carlo simulation with the task sizes

robots and the distance between the objects repository and the production area is for the left arena: 12 meters, for the middle arena 17 meters and for the right arena 23 meters. The robots are scattered initially in a robot repository

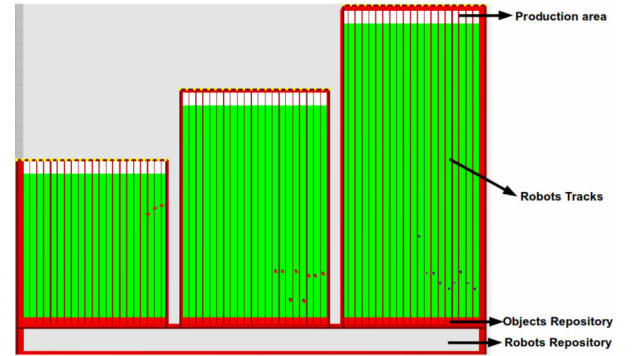


Figure 6: Multi-production scenario

tory and as soon as the task execution starts, robots use the designed probability matrices to allocated themselves to the different tasks. They start to transport the objects between their repositories and their production areas moving on separated tracks, where each track can be used by only one robot at a time. Applying the track system reduces the physical interferences between robots and allows to consider them as negligible. The only interferences present are those between the robot and its track boarders and among robots while using the robot repository area to pass to another working areas. The production areas are marked with lights to attract the robots towards them, while transporting the objects. The robots apply a *light\_attraction* behaviour combined with an *obstacle\_avoidance* while transporting the objects to their production areas and a *light\_repulsion* behaviour combined with *obstacle\_avoidance* while travelling



to fetch the objects. The simulator, <sup>1</sup>ARGoS Pinciroli et al. (2012), has been used to simulate the scenario, to measure the single robot performance and to calculate the average of the swarm performance on the 3 considered tasks. The sin-

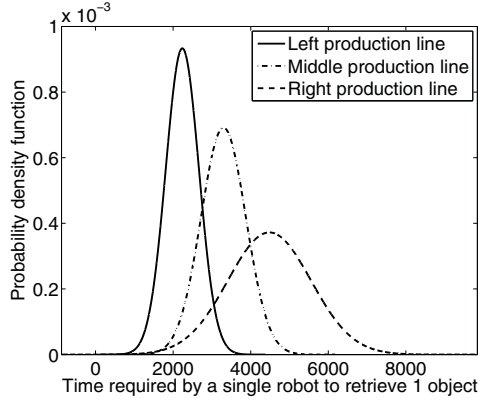


Figure 7: The probability distribution of the single robot performance on the 3 considered tasks

gle robot performance is modelled, like mentioned above, as a normally distributed random variable with a task-specific mean and a task-specific standard deviation. In our scenario, the single robot performance was measured via repeated high-level simulations in ARGoS in order to characterize the average time required by the single robot to transport one object on each of the 3 tasks. Figure 7 shows the probability density function associated with the single robot performance on each task. The measured means and standard deviations of this performance are as in following:  $\mu = \{2238, 3297.7, 4482\}$ ,  $\sigma = \{427.44, 576.69, 1071.9\}$ . The allocation strategy finds the optimal numbers to be assigned to each of the 3 tasks within their deadlines following the steps explained in section 4. The optimal number of robots to be assigned to the tasks are as following:

$$N_1 = 6 \quad N_2 = 7 \quad N_3 = 7$$

The sum of the required robots is equal to the swarm size  $N = 20$  robots, thus the need of each task will be fulfilled over all the periods where the task is active. The allocation probability matrices are calculated following Equation (13) and are as in following:

Period  $\pi_1$ :

$$\begin{pmatrix} P_1(\pi_1) & P_2(\pi_1) & P_3(\pi_1) \\ 6/20 & 7/20 & 7/20 \end{pmatrix}$$

Period  $\pi_2$ :

$$\begin{pmatrix} P_1(\pi_2) & P_2(\pi_2) & P_3(\pi_2) \\ 0 & 7/14 & 7/14 \end{pmatrix}$$

<sup>1</sup>ARGoS is a discrete-time physics-based simulation framework developed within the Swarmanoid project. It can simulate various robots at different levels of details, as well as a large set of sensors and actuators

Period  $\pi_3$ :

$$\begin{pmatrix} P_1(\pi_3) & P_2(\pi_3) & P_3(\pi_3) \\ 0 & 0 & 7/7 \end{pmatrix}$$

In this robotic example, we assume no global synchronization among the robots. Thus, we allow each robot to use the probability matrix related to the current period, each time it finishes transporting one object in order to select its next task. The fraction of robots, which is not needed in the current period, is kept idle in the robots repository.

We repeat the simulation for 10 times before calculating the average number of transported objects on each task. Figure 8 shows the comparison between the average numbers of accomplished parts on the tasks and their sizes. The small difference we can notice in Figure 8 between the number of transported parts and the number of parts required to be transported on task  $T_1$  is based on the differences of the inter-intervals of robots decisions. As robots are allowed to select their tasks each time they finish transporting one object, so the time point of the decision is based on the mean  $\mu_i$  and the standard deviation  $\sigma_i$  of the single robot performance on the considered task  $T_i$ . Hence, robots working on a specific task may be faster in taking their switching decisions than others working on other tasks, which risks keeping the robots fractions as required on all tasks over time. This is one of the weak points of allowing a dynamic switching decision each time the robot accomplishes one part of its current task. However, its effect is strongly related to the differences among the performances of single robot on the different tasks. In addition, it is necessary to use this kind of dynamic decisions, when no synchronization is available among robots to synchronize their decision points with the beginning of each period.

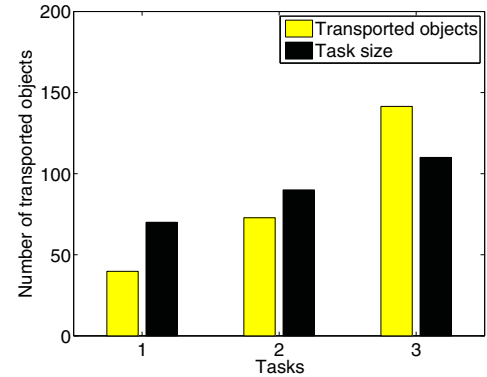


Figure 8: The Comparison between the average number of transported objects on each task and the task size

## 7 Conclusion

In this paper, we have introduced a novel task allocation strategy for swarm robotic systems in context of real-time tasks with soft deadlines. The developed strategy is a fully-

autonomous one, that uses the tasks sizes, deadlines and the single robot performance on the considered tasks to output a set of allocation probability matrices. The resulting matrices are used by the robots, independently through the execution, to allocate themselves to the different tasks with the goal of executing them within their deadlines. The considered swarm is a homogeneous one, where no communication is exploited among robots. The developed allocation strategy is a dynamic one, where robots are allowed to switch among the tasks during their execution times. This kind of dynamic allocation offers several advantages including: the possibility to cope with non-predicted lack in performance and the ability to consider on-line arrival of tasks. However it requires to assume negligible switching costs among the considered tasks. A numerical example of the allocation strategy in addition to a robotic scenario were introduced, where the allocation probability matrices were derived to be used by individual robots in a set of simulations to verify the desired swarm performance. In the future work, the impact of physical interferences on the performance on the single robot could be taken into account while estimating this performance. Considering the influence, the physical interferences has on the performance, leads to more accurate analysis of the allocation.

## References

- Acebo, E. D. and Rosa, J. L. D. L. (2008). Introducing bar systems: A class of swarm intelligence optimization algorithms.
- Agassounon, W. and Martinoli, A. (2002). Efficiency and Robustness of Threshold-Based Distributed Allocation Algorithms in Multi-Agent Systems. In *Proc. of the First Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems AAMAS-02*, pages 1090–1097.
- Agassounon, W., Martinoli, A., and Goodman, R. (2001). A scalable, distributed algorithm for allocating workers in embedded systems. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, volume 5, pages 3367–3373 vol.5.
- Bonabeau, E., Sobkowski, A., Theraulaz, G., and Deneubourg, J.-L. (1998). Adaptive task allocation inspired by a model of division of labor in social insects.
- Dahl, T. S., Mataric, M., and Sukhatme, G. S. (2009). Multi-robot task allocation through vacancy chain scheduling. *Robot. Auton. Syst.*, 57(6-7):674–687.
- Dias, M. B., Zlot, R. M., Kalra, N., and Stentz, A. (2005). Market-based multirobot coordination: A survey and analysis. Technical Report CMU-RI-TR-05-13, Robotics Institute, Pittsburgh, PA.
- Ducatelle, F., Förster, A., Caro, G. A. D., and Gambardella, L. M. (2009a). New task allocation methods for robotic swarms. In *In 9th IEEE/RAS Conference on Autonomous Robot Systems and Competitions*.
- Ducatelle, F., Förster, A., Caro, G. A. D., and Gambardella, L. M. (2009b). Task allocation in robotic swarms: new methods and comparisons. Technical report.
- Gerkey, B. and Mataric, M. J. (September 2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotic Research*.
- Gerkey, B. P. and Mataric, M. J. (2002). Sold!: Auction methods for multirobot coordination.
- Guerrero, J. and Oliver, G. (2010). A multi-robot auction method to allocate tasks with deadlines.
- Guerrero, J. and Oliver, G. (2011). Auction and swarm multi-robot task allocation algorithms in real time scenarios. In Yasuda, T., editor, *Multi-Robot Systems, Trends and Development*, pages 437–456. InTech.
- Jones, E., Dias, M. B., and Stentz, A. (2007). Learning-enhanced market-based task allocation for oversubscribed domains. In *International Conference on Intelligent Robots and Systems, 2007. IROS 2007*.
- Kalra, N. and Martinoli, A. (2006). A Comparative Study of Market-Based and Threshold-Based Task Allocation. In *Proceedings of the 8th International Symposium on Distributed Autonomous Robotic Systems (DARS)*.
- Krieger, M. J. and Billeter, J.-B. (2000). The call of duty: Self-organised task allocation in a population of up to twelve mobile robots.
- Lerman, K., Jones, C., Galstyan, A., and Mataric, M. J. (2006). Analysis of dynamic task allocation in multi-robot systems. *Int. J. Rob. Res.*, 25(3):225–241.
- Liu, W., Winfield, A. F. T., Sa, J., Chen, J., Dou, L., Liu, W., Winfield, A. F. T., Sa, J., Chen, J., and Dou, L. (2007). Towards energy optimization: Emergent task allocation in a swarm of foraging robots.
- McLurkin, J. and Yamins, D. (2005). Dynamic task assignment in robot swarms. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA.
- Nouyan, S., Ghizzioli, R., Birattari, M., and Dorigo, M. (2004). An ant-based algorithm for the dynamic task allocation problem. Technical report, IRIDIA, Université Libre de Bruxelles.
- Nouyan, S., Ghizzioli, R., Birattari, M., and Dorigo, M. (2005). An insect-based algorithm for the dynamic task allocation problem. *KI*, 19(4):25–31.
- Pinciroli, C., Trianni, V., OGrady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Caro, G., Ducatelle, F., Birattari, M., Gambardella, L., and Dorigo, M. (2012). Argos: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6:271–295.
- Schneider, J., Apfelbaum, D., Bagnell, D., and Simmons, R. (2005). Learning opportunity costs in multi-robot market based planners. In *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.
- Zhang, D., Xie, G., Yu, J., and Wang, L. (2007). Adaptive task assignment for multiple mobile robots via swarm intelligence approach. *Robotics and Autonomous Systems*, 55(7):572 – 588.
- Zheng, X., Koenig, S., and Tovey, C. (2006). Improving sequential single-item auctions. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*.

# Formal Model of Living Systems

Margareta Segerståhl

Department of Biomedical Engineering and Computational Science (BECS)

P. O. Box 12200

Aalto University, School of Science

FI-00076 Aalto, Espoo, Finland

margareta.segerstahl@aalto.fi

## Abstract

Here, a conceptual modeling formalism is proposed for the description and study of living and life-like systems. It is based on the real life and evolution of biological organisms and ultimately the system framework is determined by two functions, reproduction and survival – the two main fitness components of natural life. In the simplest form the model is a formal description of a discrete reproduction-survival state-transition system. The initial structure of the model can evolve to become more complex and it holds inherent potential for producing numerous variations of the basic theme. It is proposed, that this modeling formalism provides abstract system basis and immediately applicable conceptual tools for holistic study of living organisms on multiple levels of organization, and thereby for the understanding life and organization of complex living systems from a common systems point of view. The modeling principle is very generic and may therefore be applied directly also to the study of engineered and artificial systems.

## Systems View

The concept of an organism as a prototypic “living thing” is very intuitive for humans. But understanding life in formal terms remains a great challenge for modern science. The immense diversity and complexity of biological organisms has made it difficult to find general answers as to what the “laws of living systems” may be and how they relate to the laws of physics and chemistry. Currently there is neither commonly approved scientific vocabulary nor practical conceptual tools for forming uniform descriptions of life and living systems.

For a long time, the development of a general formalism was perhaps not considered to be an absolute necessity in biosciences. Research progressed very well also in its absence resulting in great scientific and technological breakthroughs. In the modern world, however, the concept of life is extending with increasing force to different kinds of human-made realms. Examples include topics such as artificial life, robotics, engineered minimal cells and the design of nanoscale biomolecular production plants; There are algorithmic information-based evolving systems operating in commerce, traffic, and business, only to mention some. Researchers in many different fields currently find themselves pondering what is the essence of life and living. The question is: How do

artificial and engineered systems and their properties relate to the prototypic biological organisms and natural life? How to test their “life-likeness” and living potential? How to predict the evolutionary trajectories of organism-like or life-like innovations that may reside outside the biochemistry-based internal world of cells and organisms, or even be completely information-based?

## Lists of Life

Currently, a typical way to define life is to form a list of properties that a system should have in order to be classified as living one. For example, Farmer and Belin (1990, 1991) proposed the following list of properties admitting at the same time that any such list is bound to be both imprecise and incomplete:

1. Life is a pattern in space-time, rather than a specific material object.
2. Self-reproduction.
3. Information-storage of self-representation.
4. A metabolism which converts matter and energy from the environment into the pattern and activities of the organism.
5. Functional interactions with the environment.
6. Interdependence of parts.
7. Stability under perturbations and insensitivity to small changes, allowing the organism to preserve its form and continue to function in a noisy environment.
8. The ability to evolve.

Another list (Koshland, 2002) provides seven key principles as basis for life:

1. Program.
2. Improvisation.
3. Compartmentalization.
4. Energy.
5. Regeneration.
6. Adaptability.
7. Seclusion.

These two examples already demonstrate that terminology is neither universal nor self-explanatory: No two terms are the same, yet some clearly point to the same direction. These kinds of lists also need to be accompanied by a discussion on how to interpret them and to apply them to different situations.

Overall, the lists reveal what might be the essence of the problem in terms of describing life and living systems. They are clearly complementary rather than competitive in nature. This can be seen to reflect the difficulty of making a distinction between essential and derived properties when it comes to defining the concept of life. The current situation is therefore, that *not only is life itself an enormously complex phenomenon, but also the many ways in which it is being defined and discussed in the scientific discourse* add an extra layer of conceptual complexity to the problem of dealing with it. This is far from the precision by which physical and chemical concepts can be used to describe and define phenomena in their respective fields.

Here, I take these contemplations further and propose a modeling approach that can be used to determine, define, and combine attributes of life in a more precise and orderly manner.

### Minimal Model Approach

The life and structure of living organisms can be addressed on multiple levels of organization, from molecules to ecosystems. Therefore, the task of choosing an appropriate level of observation is of central importance here – as always when modeling complex phenomena (Checkland, 2000, Flood and Carson, 1993). My solution is a two-step approach: First, only unicellular life is considered. The reason is that despite all the diversity and complexity of biological organisms, their basic constituent unit is the cell. Real unicellular organisms also demonstrate that a single cell can also be an entire organism. This starting point reduces the initial complexity of the modeling problem providing a rather precise entry point to the overall challenge of modeling organisms in general.

Then, by comparing the resulting minimal model view of unicellular life to what is generally known about real, more complex forms of cellular life, including the structure, function, and evolution of different kinds of life cycles and multicellular organisms (plants, animals, and fungi), it is possible to propose how they too can be modeled within the same principle formalism.

### System Outline

In the study of life and living organisms it is typical to focus on mature forms of actively living cells and organisms when they use energy to perform all kinds of functions. For example they can grow, move, produce many kinds of metabolites, respond to stimuli, interact and reproduce. It is also typical to assume that if an organism cannot perform these activities, it will die. However, many kinds of unicellular organisms clearly demonstrate that is not necessarily the case. Although the metabolically active reproducing form is the one that is usually studied, these organisms are often able to alter their appearance completely in order to adopt an alternative form of

existence as some kind passive inert survival structures. An example of this could be the formation of bacterial spores (Morita 1990), fungal spores, or seeds of flowering plants.

Cells of real unicellular organisms do not appear to abandon active state haphazardly. Instead the required cell-developmental changes and events are a response elicited by environmental clues that signal imminent or immediate energy deprivation (see for example the introduction in Hadany and Otto, 2007). It is typical, that once the transformation is complete the organism can withstand extreme conditions that would be destructive to the actively living form. The spore germinates when the environment again improves, and the organism returns to the active state of existence.

Based on this I propose the following minimal formalism for describing unicellular life (summarized in Figure 1). There are two states in which the conceptual living system (cell or organism) can exist. When conditions are favorable, the organism is in the *active state* and it comprises all the typical functions of metabolically active cellular life. *Reproduction* is considered to be an output function of active state living. Maintaining this state requires energy and the cell must obtain it from the environment or from its own internal energy stores. If the cell runs out of energy while it is in the active state, it will die.

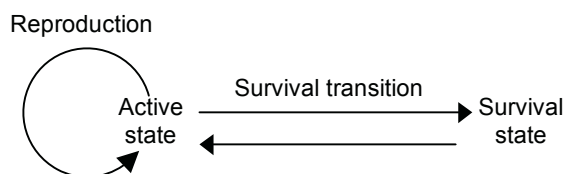


Figure 1: Minimal model formalism of living systems, based on unicellular life of real biological organisms. There are two states in which the system can exist. In the active state the system can use energy and nutrients for metabolically active living, growth, and reproduction. Formally the active state has two alternative system outputs: Reproduction (circular arrow) replicates the system without altering its state whereas survival transition takes the system into survival state. Both events require some amount of time and energy to be complete. The organism's state transition dynamics are regulated by energy availability.

The system can perform a *survival transition* when its environment turns hostile. This takes it from the active state to *survival state*. Real unicellular organisms must undergo changes in their appearance and behavior and gene expression profiles must be altered to complete this state-change successfully. In accordance with this the formal transition process assigned to require some amount of time and energy to be complete. Any additional time and energy that the system may need for returning to the active state after entering survival state simply add to the initial cost of initiating a survival transition in the first place. The minimal model survival transition is assigned to be discrete and irreversible: Once initiated, the system's only viable alternative is to enter survival state successfully.

The survival state encloses the organism's defining information content, protecting and conserving it. The



survival state can be maintained without using energy and theoretically it provides indefinite passive existence for the living system in question. The survival state is named after the only system function it provides for the organism, which is mere survival in the sense that the defining information of the organism remains in existence, albeit in an inactive form for the time being.

## The Opposite Ends of Life

In this systems formalism reproduction and survival functions define two ultimately opposite alternatives for the way in which a living organism can remain in existence. An organism that relies entirely only on reproductive active-state living corresponds to a biphasic system case where the probability for entering survival state is zero. A hypothetical example is a unicellular marine alga that exhibits no survival state in its life cycle but instead, all cells are metabolically active reproducing entities that die if active living becomes impossible. This kind of life portrays a probabilistic metapopulation-type (Fronhofer et al. 2012) life-history strategy as a combination of efficient reproduction, strong-enough dispersal of actively living cells, and a satisfactory abundance of suitable free niches at any given moment in time that the organism can potentially reach and inhabit.

An extreme case of the opposite type could be presented by an organism that has demonstrably lived, subsequently entered survival state, and then appears to remain there indefinitely thereby approaching the borderline of even being a living system anymore. A real life counterpart could be a bacterial spore or a dry seed of a flowering plant, both being what may be called individuals (of which the latter is a multicellular entity). If these structures do not exhibit any sign of active life, are they then alive or dead, living or inanimate at the time of observation?

Ultimately the “aliveness” of the organism in a situation like this cannot be determined unless we try to germinate the seed or spore. Depending on the species these kinds of structures can be extremely stable and inert, but not all of them will germinate. Not knowing beforehand what the outcome of the germination attempt would be for any one unit structure (success or failure) we may have to consider that in its survival form proper it is both alive and dead at the same time, in which case this survival stage in the system’s life presents a quantum state to the conceptual systems formalism of living system dynamics.

Interestingly, the existence of formal possibility for extremely stable survival-state occupancy makes it look like almost any kind of structural entity could be in this state and examined in the light of this living systems formalism. This could raise the issue of whether the proposed model outline is even too general, but the answer is no. It is correct that the survival state given, this model of living systems may also accommodate structurally organized states of inanimate matter. *Therefore*, being a living system comes down to the properties of the structural constitution of the individual living entity in question, its formal information storage properties, and the specific information that it contains. These features together specify the extent to which a structural organization can exhibit live behavior as determined by the biphasic reproduction-survival state transition framework.

All this results in an extended formal view of life as a system that on one side is conceptually determined purely by the survival function where it is conceptually allowed to apply also to inanimate organizations. On the ultimately opposite end of life resides pure and utter self-replication of the information-containing structure in question. This functional end provides a positive feedback loop for multiplicative existence of structural information-containing entities and possibly for their sustainable existence, formally for any kind of organization that can perform this function at a rate that is higher than the dissociation rate of the structures produced.

Not all information-containing structure in all environments have properties that enable them to live. But some clearly do, the biological organism being a prototypic example. A usual assumption in biological research is that organisms evolve to maximize reproduction. But in this model this evolutionary tendency is inherently embedded in the model formalism: There will simply be more cells that evolve towards maximizing reproductive active state living, than cells that spend excessive time in their survival state.

## Evolution In-between Reproduction and Survival

The ability to undergo adaptive evolution is a defining feature of biological organisms. Starting from the minimal-model case, the formal reproduction-survival state-transition framework may adapt and evolve to its environment in many different ways, just like a cell can undergo Darwinian adaptive phenotypic evolution with mutation and selection.

Adaptive periodic entry into survival state in a predictable fluctuating environment provides a mechanism for continuous long-term evolution towards more efficient active state living. When a cell returns from the survival state after a period of stress that would have otherwise killed it, it can continue its evolution recursively from the adaptational state where the previous round of active evolutionary living and reproduction had taken it. Sporadic entries into survival state are less likely to be equally effective in this sense, because they may take a cell into survival state also when it could alternatively undergo active state living. However, they can provide a probabilistic back-up system for organism’s survival in the case of stochastic catastrophic events that can take an actively living cell completely by surprise.

In the simplest possible minimal model scenario the survival transition may be direct and discrete, but in reality it seldom is. It appears that the cells of most real organisms spend most of their time in states where they invest simultaneously and in varying ratios to both reproduction ability and survival probability. In the system formalism they occupy *intermediate metastable transition states* that reside formally in-between the ultimately theoretical system-defining end-point states of maximum reproduction and absolute survival (see Figure 2). Because the overall evolution of a reproduction-survival state-transition system is towards active-state living, these intermediate states are likely to emerge into survival transition biology in the course of evolution as opposed to more immediate system progression in the direction of entering survival state.

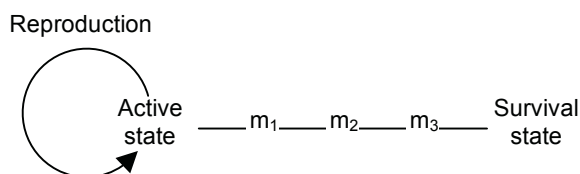


Figure 2. A simple description of intermediate metastable survival transition states (m-states) that position along the linear continuum of survival transition progression in the formal state-space of survival transition biology.

Unicellular organisms with complex life cycles can be considered from this same survival transition point of view. For example, when the yeast *Schizosaccharomyces cerevisiae* leaves the active state in order to form spores, it must proceed through several identifiable stages that provide structure to its survival transition biology. The life cycle of the unicellular malaria-causing parasite *Plasmodium falciparum* is very complex and in many respects entirely different from the yeast life cycle, but it too can be viewed from the point of view of survival transition biology. It can be seen as a stage-wise realization of a balancing act between cellular reproduction and survival functions. Flexible cell-type differentiation processes take the system from one stage to the other as this organism of reproducing cells migrates through different tissue types in its two host organisms: the mosquito and the human.

In colonial multicellularity, presented for example by some aquatic green algae, single-cell individuals form multicellular entities but each cell retains its individualistic identity and potential for future reproduction. This kind of investment to cellular properties that enable multiple cells to organize into a single functional structure can physically aid the survival of them all. Functions that allow this behavior are not directly related to the reproduction operations of a single cell, but on system level they may still contribute to this aim – especially if conditions are not ideal for single-cell living when the colony forms and the cells in question are likely to be eventually forced to enter survival state.

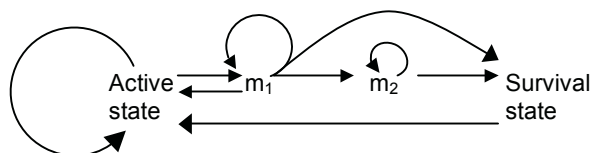


Figure 3. An arbitrary example of an evolved survival transition framework topology with metastable transition states  $m_1$  and  $m_2$ . After leaving the active state the cells of this organism may still return directly from the  $m_1$ -state. Later each cell proceeds with the transition, either directly or via another intermediate state  $m_2$  that might only be available for the cells of a supporting cell line that further protect those cells of the organism that transit directly from  $m_1$  to survival state.

### Multicellular Organisms

On one hand, complex unicellular life cycles can be formally considered to be patterns in the state-transition formalism. On

the other hand, the different cell lineages of complex multicellular organisms can each be interpreted in the same way from the same state-transition perspective regarding *both* the evolution *and* the actual developmental progression of their cellular differentiation processes.

For example, rapidly proliferating cells of an early human embryo differentiate progressively as the organism matures, their reproduction diminishes. The cells acquire features that contribute to the integrity of the developmentally complex multicellular human entity, through cell-type specific functional and structural differentiation. They adopt a role as part of a multicell-complex (a human individual) as intermediate metastable survival transition cells, and instead of entering survival state they get to live in terms of their own higher-level organizational entity in places and environments where they could never live as solitary single-cell individuals. Uncontrolled reproduction is no longer the sought-for state for the primary existence of this organism's cells, as demonstrated by cancerous cells (Merlo et. al. 2006).

Multicellular individuals are considered to be organizational entities of lower-level unit-structures. As individuals, they become subject to the same system principles of reproduction-survival state-transition dynamics as are the cells on the lower level of organization. This adds a dimension to the overall complexity of the individual organism in question. The next level of complexity expectedly arises when the multicellular individuals form social groups. A starting point for addressing this kind of conceptual *living complexity* as a formal problem in the study of complex adaptive systems in general, is to propose that the more layers of living organization there are in the formal description of an individual organism, the more complex and multilayered the interaction patterns within and between reproduction-survival state-transition dynamics on all levels of its organization.

### Conclusion

A general scientific model of life must present it as something that can be established in the realm of chemistry, obeying the laws of physics, and being manifested in the form and function of all kinds of biological organisms. The proposed modeling approach enables introduction of general systems formalism for the description and definition of very different kinds of biological systems from a common systems point of view. It should be possible to take the formal system framework of the hereby proposed model and examine it together with the key attributes of life that were listed at the beginning of this article, to see how they can best fit together and thereby deliver a concise picture of living systems in general.

In the light of this conceptual framework, *living* may refer to the things that happen in terms of this framework. *Life* on the other hand is the entire identifiable emergent overall phenomenon that arises from the operation of the living systems. More specifically, the existence of life on Earth may be seen as patterns in space-time that emerge from the structure-based functioning of a subset of information-containing systems. Based on their organizational information-containing properties, they undergo operations in their prevailing environment that *channel available energy*

(Morowitz and Smith 2007) and matter into the processes of self-maintenance and self-reproduction.

The structure of the hereby-presented reproduction-survival systems model stems from what is generally known about simple living of biological organisms. Attributes of the model are very generic suggesting, that it will be possible to use it also for testing whether other kinds of system possess the kind of formal system properties that are needed, in order to have potential for reproduction-survival dynamics and adaptive evolutionary living.

Much work is to be done in order to examine the full potential of the proposed modeling formalism and the rules by which the proposed system dynamics can operate and evolve. But the ease, by which many different versions of natural life can simply be immediately positioned even to this rough prototypic model schematic, makes this approach seem very promising. Gaining a simple and useful general formalism for the definition, description and study of living and life-like system across disciplines would be a great improvement to the current situation.

## Acknowledgements

I would like to thank Jörkki J. Hyvönen and Michael Patra for important conversations regarding the main idea of this work. I also thank Kimmo Kaski for useful discussions and for providing the necessary facilities to carry out this research.

## References

- Checkland, P. (2000) Soft systems methodology: A thirty year retrospective. *Sys. Res. Behav. Sci.* 17:S11–S58.
- Farmer, J. D. and Belin, A. d'A. (1990). Artificial life. The coming evolution. *Santa Fe Institute working paper* 90-003
- Farmer, J. D. and Belin, A. d'A. (1991). Artificial life: The coming evolution. In *Proceedings of the Second Conference on Artificial Life*, pages 815–383. After Spafford, E. H. (1995). Computer viruses as artificial life. In Langton, C. G., editor, *Artificial Life: An Overview*, pages 249–265. MIT Press, Cambridge, MA.
- Flood, R. L. and Carson, E. R. (1993) *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science*. Plenum Press, NY. 2<sup>nd</sup> edition.
- Fronhofer et al. (2012). Why are metapopulations so rare? *Ecology*. 93(8):1967–1978.
- Hadany, L. and Otto, S. (2007). The evolution of conditions-dependent Sex in the face of high costs. *Genetics* 176(3): 1713–1727.
- Koshland, D. E. Jr. (2002). The seven pillars of life. *Science* 295(5563):2215–2216.
- Merlo, L. et al. (2006). Cancer as an evolutionary and ecological process. *Nat. Rev. Cancer* 6: 924–935.
- Morita, R. (1990) The starvation-survival state of microorganisms in nature and its relationship to the bioavailable Energy. *Experientia*. 46:813–817.
- Morowitz, H. and Smith, E. (2007), Energy flow and the organization of life. *Complexity*, 13: 51–59.

# Emergence of Autonomous Behaviors of Virtual Characters through Simulated Reproduction

Yuri L. B. Nogueira<sup>1</sup>, Carlos Eduardo F. de Brito<sup>1</sup>, Creto A. Vidal<sup>1</sup> and Joaquim B. Cavalcante-Neto<sup>1</sup>

<sup>1</sup>Department of Computing, Federal University of Ceará, Brazil  
yuri@lia.ufc.br

## Abstract

This paper addresses the problem of autonomous behaviors of virtual characters. We postulate that a behavior is regarded as autonomous when the actions performed by the agent result from the interaction between its internal dynamics and the environment, rather than being externally controlled. In this work, we argue that an autonomous behavior is an agent's solution to a given problem, which is obtained through a process of self-organization of the dynamics of a system that is composed of the agent's controller, its body and the environment. That process allows the emergence of complex behaviors without any description of actions or objectives. We show a technique capable of adapting an artificial neural network to consistently control virtual Khepera-like robots by means of simulated reproduction, with no measure of the robots' fitness. All the robots are either male or female, and they are capable of evolving different kinds of behaviors according to their own characteristics, guided solely by the environment's dynamics.

## Introduction

### Contextualization

In this paper, we address the problem of autonomous behaviors of virtual characters (Shao and Terzopoulos (2007); Whiting et al. (2010)). A behavior is considered *autonomous* when the actions performed by the agent result from a close interaction between its internal dynamics and the circumstantial events in the environment, rather than from external control or specification dictated by a pre-defined plan.

That definition of autonomous behavior seems to entail an apparent contradiction to the process of creating virtual characters. Given that true autonomy implies no predefined behaviors, how is it possible to *design* the internal dynamics of an agent that is supposed to interact autonomously with its environment? The attempt to answer that question led us to investigate ways of obtaining behaviors by emergence.

Emergence can be described as the appearance of a system's global characteristic that cannot be found in any of its parts (Klaus and Mainzer (2009)). For example, although a portion of water at normal temperature is in the liquid state, we cannot say that a single water molecule can display

this property. In general, the emergent properties are associated with dynamical patterns that get established through the interactions among the component parts of the system. In our particular case, the system should be considered as composed of the agent itself, defined by a virtual body and a controller, together with every aspect of the environment, both the objects and the lawful regularities that hold in the virtual world. In this setting, we define the notion of *emergent behavior* as follows: the behavior of a virtual character is called emergent when it is not explicitly described in any of the components of the system, and arises as a result of the dynamical interaction of the components, and their specific individual properties.

It is useful to think of the emergent behavior as an agent's solution to a given problem, which is obtained through a process of *self-organization*. Indeed, the real world's biological agents constantly come up with new behaviors to overcome challenges and to adapt to a changing environment. The emergence of a new behavior reflects the process of reorganization of the internal structures of the agent. In nature, this self-organization process is controlled mainly by Darwinian evolution dynamics: generation of diversity and natural selection.

These ideas have inspired many researchers to attempt to evolve neural controllers for virtual characters using Genetic Algorithms (GA) (Sims (1994); Nogueira et al. (2008); Pilat and Jacob (2010); Palmer and Chou (2012)). So, instead of anticipating and modelling all the ways in which the agent could possibly behave, the idea is to describe a task to be achieved (that is, to create a virtual environment with challenges for survival), and let the evolutionary process shape the virtual agent's control dynamics. That is expected to lead to the emergence of behaviors, which not only solve the task, but also are coherent with the capabilities of the agent's body and with the environment's characteristics.

We argue, however, that that approach leads only to a weak form of autonomy, because the GA guides the self-organization process using a *predefined* objective function. So, in that sense, behavior is still externally described. In nature, the quality or fitness of an agent depends on its in-



ternal constitution and the way it couples with the environment. Hence, the (natural) selection criteria also constitute an emergent characteristic of the system. In this work, in order to achieve a higher level of autonomy, we present a technique for obtaining emergent autonomous behaviors of virtual characters without an externally specified objective.

### Proposed Solution

In this paper, we study the emergence of autonomous behaviors of virtual Khepera-like robots with:

- Non-interpreted simplified “vision” sensors;
- Controller consisting of an Artificial Neural Network (ANN);
- Adaptation through simulated sexual reproduction.

We show that our technique is capable of generating multiple behaviors in a population of robots: foraging, mating and obstacle avoidance. In our experiments, we could also observe different behaviors according to the gender of the robot and a complex use of the sensors for navigation.

In “Related Works” Section, we discuss the attempts of the community in obtaining autonomous behaviors of artificial agents. One notice a research trend that seeks to reduce the amount of external information provided to the system, moving from the traditional objective-driven GA to a completely environment-driven evolution, using ideas based on reproduction dynamics, similar to our work.

In “Controller” Section we describe the controller used in our virtual robots and the genetic encoding we developed to evolve it, since our simulated reproduction technique is based on the exchange of genetic material between a pair of robots of opposite genders. In “The Experiment” Section, we report the experiments, explaining the constitution of the robots and of the environment, and analyzing the dynamics of the whole system. The obtained results presented in “Behaviors” Section and final discussions are made in conclusion.

### Related Works

Evolutionary computation has long been used as a tool to develop autonomous behaviors in artificial agents (Sims (1994); Palmer and Chou (2012)). Most works address behavior as a domain specific problem, and, traditionally, have proposed solutions, which, *a priori*, fix the objectives the agents need to achieve and the metric to evaluate how well the agents perform the task of meeting the objectives. However, there are also efforts along the line of creating techniques that incorporate additional aspects of natural selection in order to obtain greater complexity and autonomy of behaviors. In this section, we will briefly discuss the research path from the explicitly objective-driven canonical genetic algorithm to environment-driven open-ended evolution (Bredeche and Montanier (2012)).

### Objective-driven evolution

The Virtual Reality community has extensively applied GAs in order to create virtual worlds automatically, in which autonomous characters present convincing behaviors. The proposed techniques are usually problem-oriented, with the evolutionary processes guided by fitness functions designed according to the expected behaviors of the characters. Some examples are the distance-based fitness of the walkers from Sims (1994) and Nogueira et al. (2008), or the speed-based fitness of the light followers from Pilat and Jacob (2010). Palmer and Chou (2012) went one step further by proposing a distributed GA that coevolves an interacting population of virtual hunter robots, instead of evolving a single individual at a time without taking into account possible interactions among them. However, the agents reproduce according to their relative fitness, based on a harvest score. The main characteristic of these works is the generation of behaviors that solve the problem in a way that is implicitly designed in the objective function.

Indeed, addressing the problem of autonomous behaviors through a problem-oriented technique, such as the canonical GA, leads to the evolution of agents capable of solving a single problem at a time. In order to achieve behavioral diversity, Schrum and Miikkulainen (2010) studied fitness-based shaping of behaviors to multiobjective domains, by dividing problems into a set of goals, i.e., a group of multiple fitness measures. A battle domain involving a scripted virtual fighter and a group of virtual monsters is used to illustrate the technique. The monsters had to maximize the inflicted damage, to minimize the received damage, and to maximize their life span. Another study that focuses on behavioral diversity is presented by Lehman and Stanley (2011), and suggests that one should abandon specific objectives and guide the search towards the novelty of solutions. These works attempt to overcome GA’s lack of behavioral diversity by proposing ways of evolving several objectives simultaneously. However, each problem an agent should solve has to be properly predefined, because it is selected through some type of performance measure.

The effects of sexual gender discrimination were also investigated through evolutionary computation. Zhang et al. (2009) proposed a GA that uses a population consisting of male and female individuals, and a fitness function based on a model of the Baldwin effect. The work is concerned with the sexual reproduction in GA, and presents numerical simulation benchmarks in order to show improvements regarding convergence speeds, prevention of premature convergence and ability to solving high dimension problems. That work incorporates another feature of natural selection to GA: the gender differentiation. However, it does not specifically analyze the effects of this new feature on the generated behavior.

Da Rold et al. (2011) studied the effects of gender determination on behavior through a simulation with male and female robots in a virtual world containing energy resources.

The reported results show that the robots acquired different patterns of behaviors according to the gender and to the pregnancy status of the females. However, the sexual dynamics was not incorporated into the evolutionary algorithm itself, since a simple GA was used, with a fitness function based on the number of matings. Mating consists in a contact between two robots of opposite genders in which the female robot gets a psychological pregnancy (i.e., it does not generate offspring), remaining in that state for a specific amount of time, during which it cannot take part in another mating. For the GA, each gender constitutes a different population, which are evolved separately, although the evaluation of the individuals depends on the interaction between the two types of robots. The exhibited behavioral diversity is related to the fact that agents with different characteristics have to solve the same problem in different ways, suggesting that gender determination is an important aspect of natural selection. However, the way this feature is exploited in that work still shows the convergence of solutions to a predefined problem.

All the works discussed so far have the common characteristic of a centralized evaluation of the agents' fitness. A paradigm shift is presented in Embodied Evolution (Watson et al. (2002)), a distributed evolutionary algorithm embodied in physical robots. In that work, the agents have a reproduction function explicitly defined in terms of their energy level, in such a way that the genes that control robots with higher energy levels have greater probability of spreading out, while those that control robots with lower energy levels have greater probability of being replaced. The environment is endowed with energy resources, and the robots that are capable of benefiting the most from them are the ones that will spread their genes. Notice that, although the robots develop a behavior that is not directly selected, one can still say that the probability associated with the reproduction function plays the role of a fitness function, because it is explicitly designed to select individuals according to the preconception that those with higher levels of energy are the fittest ones.

### Environment-driven evolution

In the environment-driven evolution approach, no fitness function is described, and the evolution is carried out by environmental pressures. That is, there is no explicit evaluation of an individual in order to select it or not, but the better performing individuals will naturally spread out according to the dynamics of the whole system.

Bredecche et al. (Bredecche and Montanier (2010); Bredecche et al. (2012)) applied this idea to evolve a population of autonomous real robots. They developed the Environment-driven Distributed Evolutionary Adaptation algorithm (EDEA), and showed that their algorithm is robust to the so called reality gap: a swarm of real robots is able to evolve efficient survival behavior strategies, with no

fitness function being ever formulated. Although their work is presented mainly from an engineering point of view, many interesting conceptual discussions arise in this context, most of them independent of particular implementations.

The authors observe that the key to EDEA is the implicit nature of fitness function, that may be seen as a result of two motivations (Bredecche and Montanier (2010)):

- **extrinsic motivation:** *agent must cope with environmental constraints in order to maximize survival, which results solely from the interaction between the agent and the environment around (...);*
- **intrinsic motivation:** *set of parameters (ie. "genome") must spread across the population to survive, which is imposed by the algorithmic nature of the evolutionary process. Therefore, genomes are naturally biased towards producing efficient mating behaviors (...).*

A low correlation between the two motivations can increase the problem's complexity, since it will possibly imply conflicting objectives. Thus, an efficient environment-driven algorithm must address a "trade-off between extrinsic and intrinsic motivations as the optimal genome should reach the point of equilibrium where genome spread is maximum (e.g. looking for mating opportunities) with regards to survival efficiency (e.g. ensuring energetic autonomy)" (Bredecche and Montanier (2010)).

The idea of environment-driven evolution fits well with our analysis of emergent autonomous behavior, since both claim that the evolution of the system should be guided by the dynamics of the interactions among its component parts. In this sense, to be precise, we can say that the evolution is not only environmentally driven, but also population-driven, or better, system-wise driven. We note that every aspect of the system may offer an opportunity for improving adaptation, in ways that cannot be foreseen *a priori*. Individual characteristics of the agents and specific behaviors cannot be judged 'good' or 'bad' in isolation, but depend on the behavior of the rest of the population, and on the current dynamics of the system. The experiment in Bredecche et al. (2012) illustrates this point well, where one can see that the individual behavior of going towards the 'sun' is 'good' (i.e., favors reproduction) because a large number of robots in the population also tend to do so. From this perspective, we can say that there is not even an implicit fixed fitness function, since the dynamics of the system may change, and so the conditions for adaptation also may change. In other words, the implicit fitness function may be considered as another emergent aspect of the system.

In this paper, we study the emergence of autonomous behavior of virtual agents, using environment-driven evolution. Simulations gave us greater flexibility and allowed us to implement robotic sexual reproduction, a feature that is still impractical to obtain in real world experiments. Therefore, we could explore additional aspects of the emergence

of autonomous behaviors, investigating, for example, the effects of population size and resources fluctuation on competitive behavior, the system's ability to follow alternative evolutionary paths, and the impact of gender differentiation on the generation of behavioral diversity.

## The Controller

### The Neural Network

The controller is essentially a Continuous Time Recurrent Neural Network (CTRNN), whose neurons are modeled in the following general form:

$$\frac{dy_i}{dt} = \frac{1}{\tau_i}(-y_i + \sum_{j=1}^n w_{ji}f(s_j) + I) \quad (1)$$

where  $t$  is time,  $y_i$  and  $\tau_i$  are, respectively, the internal state and the time constant for each neuron  $i$ ,  $w_{ji}$  is the weight of the  $j$ th input synapse of neuron  $i$ ,  $s_j$  is the state of the neuron linked to the  $j$ th input synapse,  $f()$  is the activation function of a neuron and  $I$  represents a constant input to neurons.

Furthermore, we also use two types of neurons that do not have internal dynamics: the afferent and the efferent neurons. An afferent neuron, whose internal state is the value of one of the network's input, cannot receive input from another neuron. The afferent neurons constitute the network's input layer. An efferent neuron, on the other hand, is part of the network's output layer, and its internal state is the average of the internal states of all the neurons connected to it.

### The Genetic Encoding

The controller of each robot is encoded into two chromosomes. The first chromosome encodes the stimulus  $I$  (Equation 1), and the second, which we call the Network Chromosome (NC), holds the gender of the robot and the description of the ANN itself. This grouping was chosen so that a "male brain" could evolve together with a "male gene", and a "female brain" could do so with a "female gene", while the same constant  $I$  could be tested with different networks.

The NC is defined according to a simplified version of Mattiussi's Analog Genetic Encoding (AGE) (Mattiussi and Floreano (2007)) focused at evolving a CTRNN for the control of virtual characters. To create the synapses, AGE defines an alignment score: a network-specific interaction map that leads to a complex chromosomal representation. Our proposal keeps the idea of an implicit interaction between genes that encode the synapses. However, we specify a simpler similarity function, which not only makes it possible to describe the chromosome as a simple binary array, that encodes the parameters of the network in a more straightforward way, but also maintains the advantageous properties of AGE's interaction maps for ANNs evolution.

In implicit interaction, while the neurons are explicitly described in the chromosome, the synapses are implicitly

defined, since they are formed by the interaction between genes, and not by a gene itself. To decode the ANN, we basically follow a two-step process:

1. Read the chromosome and extract the neurons and their respective input and output "ports", which we call "Neuronic Terminals" (TR);
2. Create the synapses from the interaction between the TRs.

This encoding scheme allows us to easily search augmenting topologies of neural networks.

In our work, a chromosome is an array of bits in which a single bit defines a "gender gene", and each group of 32 bits afterwards defines a regular gene. The single bit "gender gene" was introduced in order to enable sexual reproduction. The other genes are defined by a tuple  $\langle id, v \rangle$ , where  $id$  identifies the encoded element, i.e., whether it is a neuron or a TR, and  $v$  is a value that indicates a property of the encoded element.

To decode the NC, we read the first bit to determine the gender of the robot, and then we read each subsequent gene (group of 32 bits) isolating its identifier from its value. A gene identified as a neuron creates a neuron element in the network. In the decoding sequence, any TR gene that appears before the first neuron gene is ignored; and after each new neuron gene, only the first two TR genes are considered. The first of those valid TR genes determines its input terminal, while the second TR determines its output terminal. The value of a neuron specifies its time constant  $\tau_i$  (Equation 1), and the values of the TRs are used to calculate the synapses' weights between the neurons.

The first eight bits of the gene hold the  $id$  and are decoded according to Table 1. Note that the probability  $P(TR)$  is greater than  $P(N)$ , since we expect more synapses than neurons.

Id value	Meaning
$0 \leq id \leq 51$	Neuron (N)
$52 \leq id \leq 255$	Neuronic Terminal (TR)

The last 24 bits of the gene encode the value  $v$ , which is linearly mapped into a floating-point number in the range  $[-1, 1]$ . If the value is related to a neuron gene, the result is directly attributed to the time constant of the neuron. If related to a TR, it is further used to calculate a synapse weight according to the equation:

$$w(i, o) = \frac{eb \cdot (i + o)}{nb \cdot 2}, \quad (2)$$

where  $w$  is the weight of a synapse that links an output terminal of value  $o$  with an input terminal of value  $i$ . The symbol  $nb$  indicates the total number of bits that represent the value

(24 bits), and  $eb$  is the number of equal bits at the same position between the binary representations of  $i$  and  $o$ . We also defined an existence condition empirically to increase topological diversity: if  $\lfloor eb/4 \rfloor \bmod 3 = 0$  then  $w(i, o) = 0$ . The whole process of network decoding is shown in Figure 1.

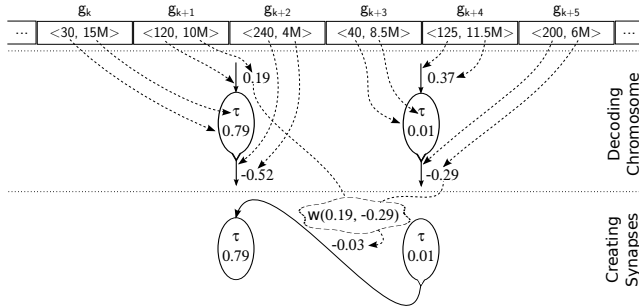


Figure 1: Building the network: First we decode the neurons and their respective terminals, then we apply Equation 2 to each pair of terminals to create the synapses. Only one synapse was created due to the existence condition (see text).

Regarding the input and output neurons, suppose that the robot has  $s$  sensors and  $m$  motors. In order to keep some structure of the ANN in the chromosome, we fix the first  $s$  genes that encode neurons to be afferent neurons, while we set the last  $m$  genes that encode neurons to be efferent neurons, that is, the inputs of the network are described in the beginning of the chromosome, the internal neurons are defined in the middle of the chromosome and the outputs are placed in the final part of the chromosome.

## The Experiment

### System description

Our simulation was developed with the *Irrlicht 3D Engine*<sup>1</sup>, with physics provided by the *Bullet Physics Engine*<sup>2</sup>. The environment is populated by simulated “male” and “female” robots that live in a square room, bounded by walls, and filled with randomly distributed fruits (Figure 2), from which the robots can get energy to live.

The robots have cylindrical bodies. A black box in the cylindrical surface represents, at the same time, the eye, the mouth and the genitals of the robot and determines its front part. The robots guide themselves through the environment using their vision, obtain energy by eating fruits and reproduce through mating. Each of these functions are better described next.

The robot’s vision is determined by three sensors positioned in the black box (the eye), as shown in Figure 3. Each sensor is able to catch the normalized distance ( $[0, 1]$ ) to the nearest object inside its “Field Of Sense” (FOS) with respect

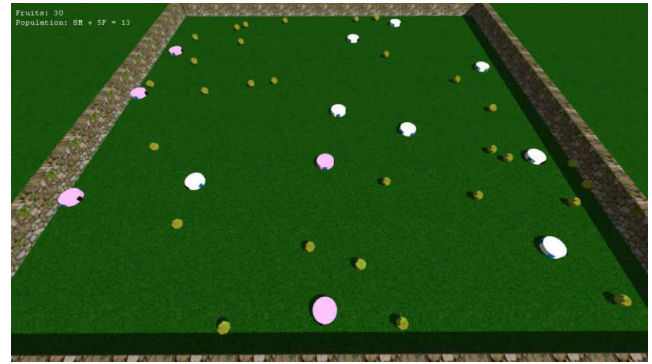


Figure 2: The environment.

to its reach (the maximum detection distance of a sensor). The sensor that is located at the center of the eye is specialized to detect walls only, and has a FOS of  $120^\circ$  and a reach of approximately  $4 * r$ , where  $r$  is the radius of the robot’s body. The other two sensors, placed at each side of the eye, are able to sense male robots, female robots and fruits, and have a FOS of  $10^\circ$  and reach of approximately  $14 * r$ . Those values were empirically chosen.

The wall sensor generates only one floating-point value indicating the normalized distance to the wall. In addition to the distance value, each one of the other two sensors also generate three bits that indicate the type of object sensed, i.e., a male robot, a female robot or a fruit. That means that the whole vision apparatus generates nine values.

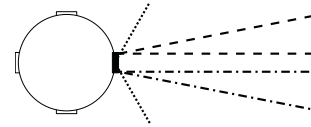


Figure 3: The distribution of the three vision sensors. The dotted lines represent the FOS of the wall sensor. The dashed lines and the dashed-dotted lines represent, respectively, the left sensor and the right sensor of robots/fruits.

Furthermore, there are proprioceptive senses of fertility and energy. The sense of fertility enables a male robot to know when it is infertile (1 if infertile, 0 otherwise) and a female robot to know when it is pregnant (1 if pregnant, 0 otherwise). The sense of energy enables a robot to know its level of energy, which ranges from 0 (the robot is fully energized) to 1 (the robot is totally exhausted). Therefore, the strength of the signal allows the robot to perceive when its energy is finishing. Thus, there are nine signals of vision and two signals of proprioceptive senses, leading to an ANN with eleven afferent neurons.

A robot has two motors, which are controlled by two efferent neurons respectively. When the first motor receives a signal from its efferent neuron, it moves the robot forward in case the value of the signal is positive, and moves the robot

<sup>1</sup><http://irrlicht.sourceforge.net/>

<sup>2</sup><http://bulletphysics.org/>



backward if the signal is negative. Likewise, the other motor makes the robot turn right in the event of a positive signal, and makes the robot turn left if the received signal is negative. The actions of the motors are simplified and are not physically accurate. The amplitude of the signals generated by the efferent neurons are increased 100 times, before they are applied as the robot's speed.

The energy level of a robot increases 7,500 energy units (eu) whenever a fruit is eaten, i.e., when the robot touches a fruit with its mouth. The maximum energy value is 100,000eu and continuously decreases in direct proportion to the applied motor signals plus a value proportional to the robot's age. If the energy is exhausted, the robot dies and is removed from the environment. Thus, for example, suppose that  $o_1$  and  $o_2$  are the amplified output of the two efferent neurons of the ANN and that  $t$  is the robot's age. Then, the energy consumption  $C$  is calculated according to the equation:

$$C = (|o_1| + |o_2|)^2 + \frac{t}{75.0}. \quad (3)$$

Regarding the dynamics of fruits replacement, at each time step, a new fruit is randomly placed in the room, provided that:

- The number of fruits does not exceed 30 fruits; and
- The total number of objects, i.e., the number of fruits plus the number of robots, does not exceed the limit of 45 objects.

### Simulated Reproduction

The simulation starts with 15 robots. If the population becomes smaller than 6 individuals, we place 15 new random individuals in random positions. Each of these robots has its energy randomly initialized with a value between 10,000eu and 20,000eu. The genetic information is also randomly generated. Since we have just one bit to express the robot's gender, 50% of the population consists of female robots.

Mating is consummated whenever a male robot's genitals (the black box) touch the body (any part of the cylinder) of a fertile female robot. A female is fertile if its energy is greater than 25,000eu. Since any robot has an initial energy of 20,000eu at most, every female is infertile at first, and needs to eat some fruits in order to reproduce.

If mating occurs, the male robot gets infertile during 250 simulation steps and the female robot gets pregnant. The new robot is placed adjacent to its mother, so the female robots remains pregnant until she goes to a free place where its child can be positioned. The child's energy is initialized with a value between 15,000eu and 25,000eu, which is taken from its mother. Therefore, when, at the moment it gets pregnant, a female robot has low energy (a value close to 25,000eu), there is a higher probability that it will die sooner. After giving birth, the female gets infertile during 250 simulation steps to avoid a pregnancy right after the other.

The chromosomes of the new individual are generated by crossing over the parents' chromosomes. Since a chromosome is simply an array of bits, the process of crossover sets two breakpoints randomly, and exchange the bits between the pair of chromosomes at the defined range. Note that this method can generate mutation by breaking a gene, since it is defined by a group of 32 bits. This is expected to create variability. We also apply an explicit mutation, randomly changing bits in the chromosomes with a probability of 0.1%. Since after the crossover we still have a pair of chromosomes, we simply discard one randomly.

The genetic information of an individual encodes its ANN directly. Therefore, when crossover takes place between a pair of chromosomes of different individuals, the process can be viewed as if pieces of each individual's brains were being exchanged. Consequently, the newly generated brain can lead to a robot with behavioral traits inherited from both parents.

### Life Dynamics

Note that, according to the reproduction dynamics described, if there is a high density of fruits and robots at the environment, mating is relatively easy to occur and can occasionally happen in a random way. In fact, this is necessary to avoid endless resumptions of the population and to bring some line of evolution. In so far as that the population evolves, those individuals who present some type of strategy are at a greater advantage and will impose new conditions to the system, causing random behaviors to decrease.

Another important point to comment is the balance between the number of individuals in the population and the amount of energy resources. According to the fruits' replacement dynamics, if the population grows, the number of fruits available reduces. Therefore, with scarce energy resources in the environment, the robots with worse performance will die. Note that, if the population size grows above 45, no fruit will appear and, thus, the robots that are less efficient will die before new fruits appear. That dynamics prevents population explosion automatically and provides some selective pressure, which guides evolution.

### Behaviors

We ran the simulation several times and obtained mixed interesting behaviors. Due to space constraints, we cannot present the results of all the runs<sup>3</sup>. Therefore, we will focus the discussion in a common recurring result: females tend to seek food and males tend to look for mating. Furthermore, the robots learned how to deviate from the wall and how to use their simple vision to guide themselves through the environment efficiently and meet their needs. In Figure 4, a sequence of frames shows two robots presenting the mentioned behaviors, which are detailed in the respective labels.

<sup>3</sup>Watch the video with multiple runs of our experiments: <https://www.youtube.com/watch?v=zyDdjD6d5CE>

Other interesting strategies also emerged from the population in different simulation runs, always leading to the population's survival and stability.

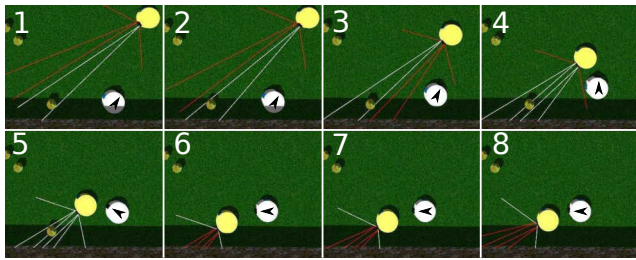


Figure 4: Commonly observed behaviors. Note that the male robot (the one with the arrow indicating his direction) follows the female robot that seeks the fruit. Another interesting point to emphasize, is the complex use of the simple vision: the female senses the fruit with her left sensor (Frame 1), and then turns left (Frames 2 and 3) in order to use the right sensor to determine the direction to follow, correcting the movements and maintaining the object between the two sensors (Frames 4 and 5). Frames 6, 7 and 8 show the female robot turning right in order to deviate from the wall after catching the fruit.

Since we do not have to define any objective, there is no variable to watch in order to follow convergence toward an expected behavior. However, some values indicate the evolution of the whole population, and the analysis of the relations among those values allows us to objectively demonstrate the emergence of the described behaviors.

The mean lifespan of the population over a period of time is a good parameter to see the emergence of some strategy of the population in order to survive. Another aspect that can indicate characteristics of the behaviors is the correlation between the male and female lifespans. Figure 5 shows the mean lifespans of the population in the simulation run described in Figure 4. Note that the female robots have converged to a mean lifespan greater than that of male robots. This is related to the fact that the female robots were always searching for food actively, while the male robots caught a fruit occasionally. However, both gender increased their efficiencies.

The average number of collected fruits and the average number of matings are good parameters to analyze the behavioral characteristics of each gender. In Figure 6a we can see that, in the analyzed simulation run, the female robots converged to collect about 10 fruit on average during their lifespan, while the average number of fruits collected by the male robots is less than one, demonstrating the preference of female robots for collecting fruits. In Figure 6b, we can note the preference of the male robots for the mating behavior. However, it is important to mention that other runs presented different strategies, such as, for example, the for-

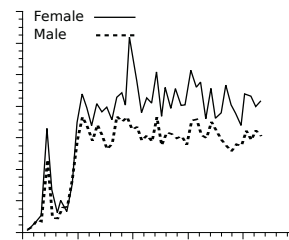


Figure 5: Mean lifespan of the population every 900 seconds of simulation.

mation of clusters of robots, which increase the probability of matings, and the presence of robots with both foraging and mating behaviors, regardless of the gender.

The observation of the population size at a given time along with the average number of collected fruits and the mean lifespan, shows some aspects of the general behavior of the whole population. Note that there is a peak in the graph of Figure 6a before convergence around a certain smaller value. Analyzing Figures 5 and 6a at approximately the same time (about 4000 seconds and 1 hour, respectively), we can see that as the robots learn how to catch fruits, there is an increase in the mean lifespan. As described, the number of fruits placed in the environment depends on the number of robots. So, an increase in the mean lifespan leads to population growth and, consequently, to reduction of the available resources, hence reducing the average number of collected fruits per robot. That leads to a balance of the population size, preventing population explosion, as shown in Figure 7.

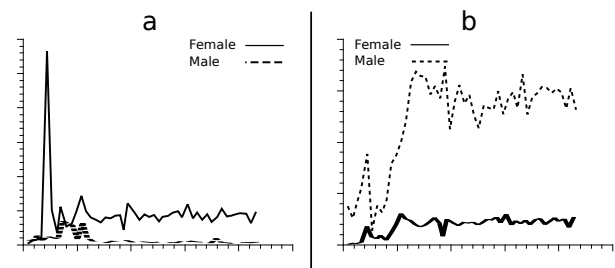


Figure 6: Average number of collected fruits and matings every 900 seconds of simulation. (a) The greater number of fruits collected by the female robots indicates their behavioral tendency to foraging. (b) Note the increase of matings with time. This shows a male preference to such behavior.

## Conclusions

We described an artificial life system where virtual Khepera-like robots developed multiple autonomous behaviors, without any description of their objectives. The observed behaviors emerged solely from the self-organization of the dynam-

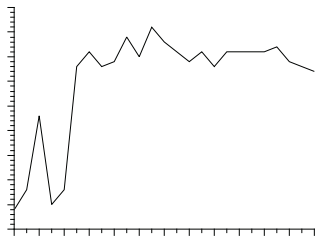


Figure 7: Population size every 30 minutes. Note that with 2.5 hours of simulation, the population size increases to about 45 individuals and then is balanced around this number due to shortage of resources caused by the large amount of robots that developed the foraging behavior.

ics of the system. The robots were divided into genders and were controlled by an ANN. We presented a genetic encoding for the ANN, which allowed the adaptation of controllers through simulated reproduction, providing an implementation of environment-driven evolution.

The system was capable of exhibiting several types of behaviors, according to the robots' characteristics. A common situation observed, was the emergence of mating behavior in male robots and foraging behavior in female robots. A single individual was also able to show multiple behaviors, such as avoiding collisions with the walls and use of vision to pursue its own objectives. Although different behaviors have emerged from different simulation runs, the system was always able to show the evolution of robots presenting strategies that led to an increase in the mean lifespan and in the size of the population.

The results of our experiments show that the self-organization of the system is capable of producing an intimate coupling between agent and environment, producing complex and natural behaviors without any *a priori* description. This characteristic is clearly illustrated with the strategy developed by the virtual agent to compensate for its primitive visual sensory apparatus and to be able to find a direction to an object, an information that is not originally provided by the sensors.

### Acknowledgements

This work was supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

### References

- Bredeche, N. and Montanier, J.-M. (2010). Environment-driven Embodied Evolution in a Population of Autonomous Agents. In *Parallel Problem Solving From Nature*, pages 290–299.
- Bredeche, N. and Montanier, J.-M. (2012). Environment-driven Open-ended Evolution with a Population of Autonomous Robots. In *Evolving Physical Systems Workshop*, East Lansing, United States.
- Bredeche, N., Montanier, J.-M., Liu, W., and Winfield, A. (2012). Environment-driven Distributed Evolutionary Adaptation in a Population of Autonomous Robotic Agents. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):101–129.
- Da Rold, F., Petrosino, G., and Parisi, D. (2011). Male and female robots. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 19(5):317–334.
- Klaus and Mainzer (2009). From embodied mind to embodied robotics: Humanities and system theoretical aspects. *Journal of Physiology-Paris*, 103(3-5):296 – 304.
- Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evol. Comput.*, 19(2):189–223.
- Mattiussi, C. and Floreano, D. (2007). Analog Genetic Encoding for the Evolution of Circuits and Networks. *IEEE Trans. Evol. Comput.*, 11(5):596–607.
- Nogueira, Y. L. B., Vidal, C. A., and Cavalcante-Neto, J. B. (2008). A nervous system model for direct dynamics animation control based on evolutionary computation. In *SAC '08: Proc. of the 2008 ACM symposium on Applied computing*, pages 1793–1800. ACM.
- Palmer, M. E. and Chou, A. K. (2012). An artificial visual cortex drives behavioral evolution in co-evolved predator and prey robots. In *Proc. of the 14th intl. conf. on Genetic and evolutionary computation conference companion*, GECCO Companion '12, pages 361–364. ACM.
- Pilat, M. L. and Jacob, C. (2010). Evolution of vision capabilities in embodied virtual creatures. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO '10, pages 95–102, New York, NY, USA. ACM.
- Schrum, J. and Miikkulainen, R. (2010). Evolving agent behavior in multiobjective domains using fitness-based shaping. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO '10, pages 439–446, New York, NY, USA. ACM.
- Shao, W. and Terzopoulos, D. (2007). Autonomous pedestrians. *Graph. Models*, 69(5-6):246–274.
- Sims, K. (1994). Evolving virtual creatures. In *Proc. of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH 1994, pages 15–22. ACM.
- Watson, R. A., Ficici, S. G., and Pollack, J. B. (2002). Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18.
- Whiting, J. S., Dinerstein, J., Egbert, P. K., and Ventura, D. (2010). Cognitive and behavioral model ensembles for autonomous virtual characters. *Comput. Intell.*, 26(2):142–159.
- Zhang, M., Zhao, S., and Wang, X. (2009). A hybrid self-adaptive genetic algorithm based on sexual reproduction and Baldwin effect for global optimization. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, CEC'09, pages 3087–3094, Piscataway, NJ, USA. IEEE Press.

# Robust Optimization of Adjustable Control Factors Using Particle Swarm Optimization

Takeo Kato<sup>1</sup>, Koichiro Sato<sup>2</sup> and Yoshiyuki Matsuoka<sup>2</sup>

<sup>1</sup>Department of Mechanical Engineering, Tokai University

<sup>2</sup>Department of Mechanical Engineering, Keio University  
t.kato@tokai-u.jp

## Abstract

Most conventional robust design methods assume design solutions are fixed values. Using these methods, designers set each control factor to a fixed value to maximize the robustness of objective characteristics. However, fluctuations in the objective characteristic often exceed the allowable range in a design problem. Consequently, obtaining sufficient robustness is difficult using conventional methods.

This research defines adjustable control factors whose values can be adjusted within a given range to increase robustness and proposes a method to calculate robustness, including factors to adjust the objective characteristic and to derive optimum ranges of the factors. The robustness index, which indicates the feasibility that the objective characteristic values are within the tolerance by the adjustment, is calculated by the Monte Carlo method, while the range of adjustable control factors is optimized using the Vector evaluated particle swarm optimization. Finally, an engineering example is presented to demonstrate the applicability of the proposed method.

## Introduction

Robust design aims to ensure product performance robustness against fluctuant factors, such as user characteristics and material properties, by deriving the optimum (unique) value of the design parameter (design solution). Due to globalized markets and material procurement, robust design has received much attention, and many robust design methods have been proposed (Matsuoka 2010). Some methods evaluate robustness of the objective characteristic using an orthogonal array for efficiency (Sundaresan et al. 1991; Taguchi, 1993; Yu and Ishii, 1998ab), while others derive robustness using the objective characteristic values calculated via a Taylor series approximation (Arakawa and Yamakawa, 1995; Belegundu and Zhang, 1992; Emch and Parkinson 1994; Parkinson et al. 1993, 1995; Ramakrishnan and Rao, 1996; Zhu and Ting, 2001). Additionally robustness has been calculated as the feasibility of the objective characteristic being within the tolerance to consider the objective characteristic distribution (Eggert and Mayne, 1993; Watai et al. 2009).

In most conventional methods, designers set control factors to fixed values to maximize the robustness. In cases where the objective characteristic distribution is smaller than the tolerance (Figure 1a), these methods can derive a design solution (optimized control factor values)  $x_0$  with sufficient

robustness. However, in cases where the objective characteristic distribution is larger than the tolerance (Figure 1b), a solution to sufficiently maximize robustness cannot be obtained. In such cases, the control factors must be adjusted to ensure robustness. In other words, as the values of the control factors are varied, the whole of the objective characteristic distribution should be located within the tolerance (Figure 1c).

The concept of adjusting the factors originates from Taguchi's method (Taguchi, 1993). In this method, the control factors are set to minimize the objective characteristic fluctuation, and then the designer selects a tuning factor, which has a negligible effect on the fluctuation, to minimize the difference between the nominal value of the objective characteristic and its target value. Otto (Otto and Antonsson, 1993) assumed the tuning factor is adjusted after the fluctuation of objective characteristic. Hence, after the objective characteristic fluctuates, the factors are altered to minimize the difference between the fluctuated objective characteristic values and their target values. Otto proposed a method to evaluate the robustness using the expected value of the objective characteristic adjusted by the tuning factors. This method provided a new concept of robustness in which an adjustment improves the robustness and it helps relax the design requirements such as dimension tolerance and material property tolerance. However, Otto's method is not applicable to design problems where the designer chooses the tuning factors and their adjustable ranges because these parameters are preliminarily set in Otto's method. This is a common problem in mechanical design. For example, to design a seat, the designer must determine which adjustable mechanism, such as seat reclining mechanism, and its adjustable range to apply to the seat. Because a method to address such a design problem has yet to be proposed, the designer must determine these parameters using his/her personal design experience.

This research proposes a method to derive the optimum range for the adjustable factors which are chosen adequately to improve the robustness of the objective characteristics. This paper is divided as follows. Section 2 presents definitions and terminologies. The proposed robustness index, calculation method, and range optimization of the factors using the Vector evaluated particle swarm optimization (VEPSO) are described in Section 3. Section 4 illustrates an application of the proposed method to a seat design problem, while Section 5 provides conclusions and the future research direction.



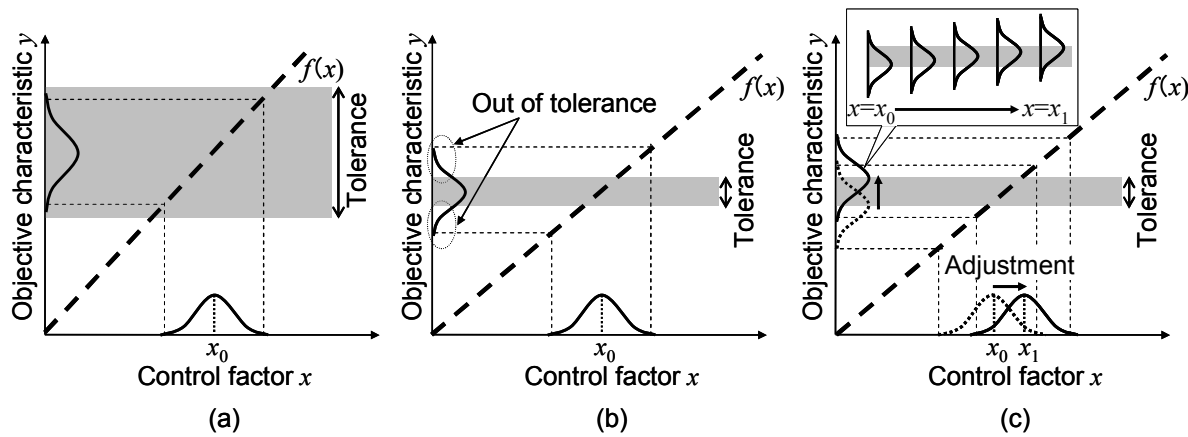


Figure 1: Conceptual illustration of a design problem that includes an factor whose value is adjustable elements)

## Definitions and terminologies

In robust design, objective characteristic (product performance)  $y$  fluctuates according to fluctuant factors (control factors  $x$  and noise factors  $z$ ). Although the values of control factors fluctuate, designers can set their nominal values, but not those of noise factors. In this paper, adjustable control factors (hereinafter called ACFs) whose values  $t$  can be adjusted in the adjustable range  $[t_l, t_u]$  anytime while using or manufacturing the product to maintain the objective characteristic are newly defined.  $t_l$  and  $t_u$  are the lower and upper values of ACFs, respectively. ACFs are similar to the tuning factors defined in Otto's method (Otto and Antonsson, 1993) with respect to their adjustment, but the adjustable ranges of ACFs and tuning factors differ (i.e., the designer defines the ranges of the ACFs).

The concept of robustness in this research is defined below. If the tolerance of objective characteristic  $[y_l, y_u]$  exists as shown in Figure 1, then ACFs can be adjusted to locate each fluctuation value of the objective characteristic within the tolerance. Hence, the robustness index for ACFs ( $R_A$ ) is defined as the feasibility that the objective characteristic values are within the tolerance at least once, by the adjustment of ACFs. Using ACFs and  $R_A$ , design problems in this research are expressed as:

$$\begin{aligned} &\text{Find } [t_l, t_u], x \\ &\text{to maximize } R_A( y(= f(x, z, t)) ) \\ &\text{to minimize } |t_u - t_l| \end{aligned} \quad (1)$$

where  $f$  is the objective function. To prevent an unnecessary expansion of the ACF ranges, which increases the production costs and failure rate, this formulation does not only maximize  $R_A$ . Minimizing the size of the range described in Equation (1) is an example of preventing an unnecessary expansion because other factors (e.g., the form and location of the range) can lead to the aforementioned issues.

Below are definitions and descriptions of the terminologies used in this paper.

*Objective characteristic* ( $y = f(x, z, t)$ ): The characteristic to express the function of the design objective, and is calculated by objective function  $f$ .

*Control factors* ( $x = \{x_i\}, i=1, 2, \dots, n_x$ ): Factors whose nominal values are set by the designer, but fluctuate the objective characteristic.  $n_x$  represents the total number of control factors.

*Noise factors* ( $z = \{z_i\}, i=1, 2, \dots, n_z$ ): Factors that fluctuate the objective characteristic, but their nominal values cannot be set by designers.  $n_z$  expresses the total number of noise factors.

*ACFs* ( $t = \{t_i\}, i=1, 2, \dots, n_t$ ): Control factors with nominal values that can be adjusted within their adjustable ranges.  $n_t$  denotes the total number of ACFs.

*Adjustable range of ACFs* ( $[t_u, t_l]$ ): The range, determined by the designer, where the ACFs are adjustable.

*Assignable points of ACFs* ( $\{t_j\}, j=1, 2, \dots, n_{ap}$ ): The combinations of ACFs' values that can be varied to be within the adjustable range.  $n_{ap}$  denotes the number of ACFs' assignable points.

*Robustness index* ( $R_A$ ): Index to evaluate the robustness of the objective characteristics with regards to the ACFs adjustment.

## Robust design method for ACFs

### A robustness index for ACFs

In conventional robust design methods, robustness indices are approximated to improve the calculated efficiency. For example, the index in Ramakrishnan's method (Ramakrishnan and Rao, 1996), which is the weighted sum of the mean value and standard deviation of the objective characteristic, is calculated using a Taylor series approximation. However, approximated values significantly differ from the actual values or cannot be derived in the cases where the followings are not satisfied: (1) the objective characteristics monotonically increase or decrease with respect to the factors, (2) the objective function is differentiable, (3) the fluctuations in the factors are sufficiently small, and (4) the factors are independent from each other.

The proposed robustness index must be calculated accurately because the adjustable range must be minimized, as shown in

Equation (1). In other words, using an accurate robustness index, the designers must set the adjustable range as small as possible. Consequently, the Monte Carlo method, which derives accurate values but is time consuming, is applied to calculate the index. The calculation methods is described below.

All ACFs' assignable point values should be used to calculate  $R_A$ . Specifically, sets of objective characteristic fluctuations that satisfy the tolerance are derived with respect to all assignable point values  $t_j$ , as shown in Figure 2.  $R_A$  is calculated as the ratio of the sum of the sets of fluctuant combinations of  $x$  and  $z$  where at least one of the objective characteristic values  $y_j$  derived from  $t_j$  is within the tolerance as shown in Equation (2).

$$R_A = P \left[ \bigcup_{j=1}^n \{ C(x, z) | y_1 \leq f(x, z, t_j) \leq y_u \} \right], \quad (2)$$

where the square bracket expresses a set of  $C(x, z)$  where the objective characteristic is located within the tolerance by adjusting ACFs. This means  $R_A$  is the rate of the set and the entire set. The assignable point values are expressed as a finite number of discontinuous values  $t_j$  because  $R_A$  is calculated using the Monte Carlo method. The number of the assignable (discontinuous) values should be sufficient to assume the ACF is continuous. However, the number should be decreased if the calculation amount is too large. To calculate  $R_A$ , first,  $s$  random combinations of the control and noise factors are generated based on their probability density functions. Second, objective characteristic  $y_i$  is calculated using the generated random combinations  $\{x_i, z_i\}$  ( $i=1, 2, \dots, s$ ) and all the assignable point values. That is, the number of calculating objective characteristic values is the product of the random combination number  $s$  and the assignable points numbers of ACFs. Finally, the values calculated from each random combinations of  $x_i$  and  $z_i$  are assessed to determine whether at least one of the calculated values is within the tolerance (i.e., at least one assignable point which consists an objective characteristic value that satisfies the tolerance). Then  $R_A$  is calculated as:

$$R_A = \frac{1}{s} \sum_{i=1}^s M_i$$

$$\left( M_i = \begin{cases} 1 & (\exists t \in \{t_j\}; y_1 \leq f(x_i, z_i, t) \leq y_u) \\ 0 & (\text{otherwise}) \end{cases} \right) \quad (3)$$

$$j = 1, 2, \dots, n_{ap}$$

### Adjustable range optimization

This study proposed a optimization algorithm using the VEPSO in order to solve the design problem of Equation 1. An outline of the VEPSO and the algorithm using it are described below.

**Outline of VEPSO.** The VEPSO (Vlachogiannis and Lee, 2005, 2009) is an improved method of the PSO (Kennedy and Eberhart, 1995) that is one of the representative

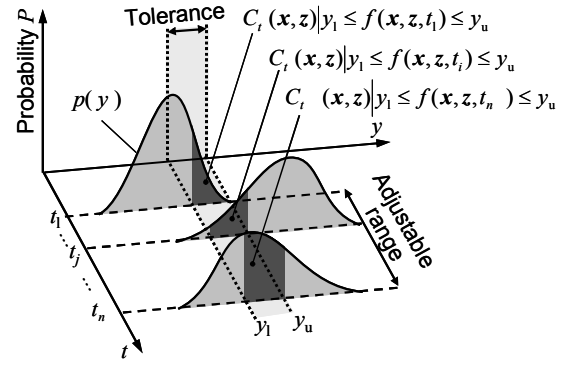


Figure 2: Set of the objective characteristic fluctuations used to calculate the robustness index

metaheuristics, in order to handle the multi objective optimization problems. The PSO imitates the movement of organisms in a bird flock or fish school and searches a solution using the information both from the individuals (particles) and their swarm. The VEPSO assigns an objective to each of swarms and searches a solution using the information inside or between swarms. The location vector (i.e. design variables) of the  $i$  th particle in the  $j$  th swarm  $x_i^{[j]}$  is updated as follows:

$$x_i^{[j]}(T+1) = x_i^{[j]}(T) + v_i^{[j]}(T), \quad (4)$$

where,  $T$  is the number of iterations.  $v$  is the velocity vector to direct the particles to the updated locations and is calculated as:

$$v_i^{[j]}(T+1) = k [ w v_i^{[j]}(T) + c_1 r_1 \{ x_{pb,i}^{[j]}(T) - x_i^{[j]}(T) \} + c_2 r_2 \{ x_{gb,i}^{[s]}(T) - x_i^{[j]}(T) \} ]$$

$$\left( s = \begin{cases} M & \text{if } j = 1, \\ j-1 & \text{if } j = 2, 3, \dots, M \end{cases} \right), \quad (5)$$

where,  $M$  is the number of swarms;  $c_1$  and  $c_2$  are the parameters to express the degree of incidence of the private best location of each particle  $x_{pb}$  and the global best location  $x_{gb}$ , respectively;  $r_1$  and  $r_2$  denote the random numbers uniformly distributed in  $[0, 1]$ .  $w$  is the parameter to define the effect of the current velocity vector and decreases based on  $T$  as shown in the following equation:

$$w(T) = w_{\max} - \frac{w_{\max} - w_{\min}}{T_{\max}} \cdot T, \quad (6)$$

where,  $w_{\max}$  and  $w_{\min}$  are the maximum and minimum value of  $w$ .  $T_{\max}$  is the maximum number of the iterations.  $k$  denotes the parameter relating the convergence performance and expressed as follows:

$$k = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \quad (\varphi = c_1 + c_2). \quad (7)$$

As shown in Equation (5), the velocity vectors are defined using the global best locations of the different swarms. This enables the solution search based on the information from the other swarms and the global locations (solutions) of the swarms to approach each other. Therefore, the solution search

of the VEPSO has the following features: assigning a objective to each of swarms and searching the solutions located close to each other. These features generate the following merits to solve the robust design problem (Equation (1)): 1) the design solution (assignable points) to ensure the robustness can be efficiently derived by assigning the robustness regarding the part of the fluctuation of the factors to each of the swarm; 2) the distance between assignable points (the adjustable range) can be smaller.

**Procedure to derive optimum adjustable range using VEPSO.** This study amended the robustness index (Equation (2)) in order to evaluate each assignable point that assures the robustness regarding the part of the fluctuation of the factors. The robustness assured by the  $j$  th assignable point is expressed as follows:

$$R_j = P[C(x, z)_j | y_l \leq f(x, z, t_j) \leq y_u], \quad (8)$$

where,  $C(x, z)_j$  ( $\in C(x, z)$ ) is a part of the fluctuation of the factors assigned to the  $j$  th assignable point and holds the following equation:

$$\sum_j C(x, z)_j = C(x, z). \quad (9)$$

When considering the four assignable points, the four swarms are defined and search for the adjustable range (assignable points) based on the VEPSO procedure. The optimization algorithm using the VEPSO is described in Figure 3. In this algorithm, the parameters of the VEPSO (e.g.  $c$ ,  $w$ ,  $T_{\max}$ , etc) are firstly set. Next, the number of the assignable points is decided and the same number of the swarms are set. The locations of the particles are updated based on the objective (robustness)  $R_j$  ( $j=1, 2, \dots, n_{ad}$ ). The update of the locations iterates until  $T = T_{\max}$ , and the global best location of the swarms are derived as an design solution (adjustable range).

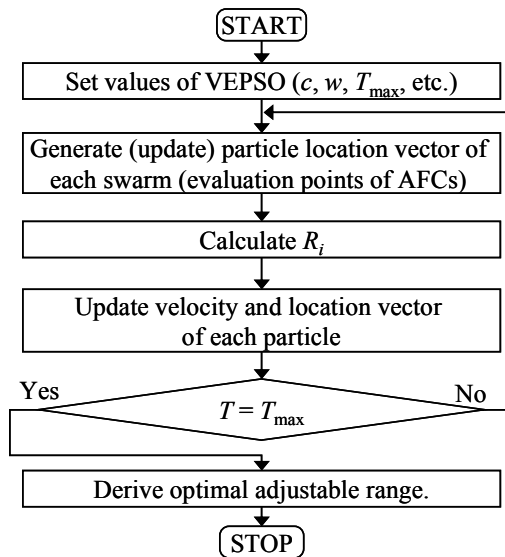


Figure 3: Proposed algorithm of robust design method

## Illustrative example

### Problem description

To demonstrate the proposed robust design method, we applied it to a seat design for railway vehicles because numerous people with diverse physiques and sitting postures use these seats. However, the conventional seat design typically assumes an average physique and posture. Thus, designing a seat that is robust for various physiques and postures is desirable. Herein the design objective focused on the hip-sliding force, which is generated on the buttocks by the static instability of the upper and lower body masses, causing discomfort when sitting (Matsuoka 2000). Therefore, the design objective is to inhibit the hip-sliding force for various physiques and postures.

Table 1 defines the objective characteristic and factors of this design. The control factors, seat cushion angle (hereafter called C.A.), seat back angle (B.A.), and forward tilt angle of the upper seat back (F.A.), can be adjusted by the mechanisms for the seat cushion forefront lifting function, reclining function, and forward tilt function, respectively. Previous research (Matsuoka 1988) has demonstrated the influence of these angles on the hip-sliding force. Therefore, these angles are considered AFCs. Noise factors include users' physiques and sitting postures. The physiques are defined based on actual measurements of Japanese citizens (National Institute of Bioscience and Human Technology, 1996). Additionally, we considered three sitting postures: a standard sitting posture where the lumbar region is in contact with the seat back, a stretched waist sitting posture where the waist is stretched and slid forward from the standard sitting posture, and a bent waist sitting posture where the waist is bent and slid forward from the standard sitting posture. The ratio of these sitting postures is 3:1:6 (Matsuoka, 2000).

Modeling the sagittal plane of the human body and the seat was used to derive the objective function (Figure 4). The human model assumes that the movements of the low momentum joints are zero and is consequently, composed of four high momentum joints. On the other hand, the seat model, constructed based on the existing seat found on a 485 train (Hatsukari), is divided into three parts: seat cushion, upper seat back, and lower seat back, which are rigidly linked. Figure 5 shows the objective functions (the formula to compute the hip-sliding force for each posture) derived based on these models. The robustness index ( $R_A$ ) is derived as the weighted sum of the indices calculated using these formulae.

Additionally, in the PSO, the parameters (e.g.  $c$ ,  $w$ , etc) are important for the convergence or the computational efficiency. Therefore, this study implemented some optimizations regarding the recommended values of the parameters in the conventional studies and compared the results to clarify the proper values of them. This study focused both on  $c_1$  and  $c_2$  and conducted the four analyses using the parameter combinations: ( $c_1=2.80$ ,  $c_2=1.30$ ) recommended by Carlisle (Carlisle and Dozier, 2001); (2.05, 2.05) suggested by Kennedy (Kennedy, 1998); (1.55, 2.55) and (1.05, 3.05) that are smaller values of  $c_1$ . This study also implemented the optimization using the traditional genetic algorithm (GA). The definition of the parameters is summarized in Table 2.

Items	Definition
Hip-sliding force (as objective characteristic)	Tolerance of Hip-sliding force: -10 to 20 N.
Seat cushion angle $\theta_C$ Seat back angle $\theta_B$ Forward tilt angle $\theta_F$ (as ACFs)	$10 \leq \theta_C \leq 25$ $20 \leq \theta_B \leq 35$ , $\theta_C + 10 \leq \theta_B$ $0 \leq \theta_F \leq 30$
Body height $L$ Body height $M$ Sitting posture (as noise factors)	$L$ and $M$ are normal distributions Mean value of $L$ : 1.65m, standard deviation of $L$ : 0.08m Mean value of $M$ : 58.1kg, standard deviation of $M$ : 9.09kg Ratio of standard, stretched waist, and bent waist sitting postures: 3:1:6

Table 1: Definition of objective characteristic and factors

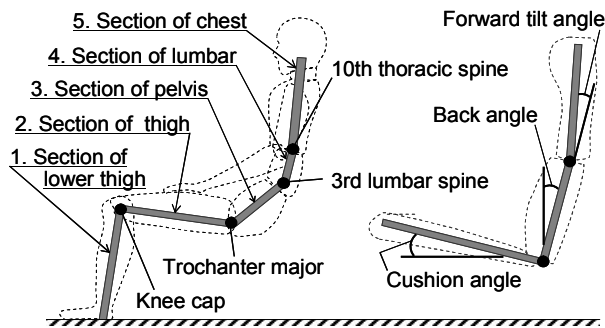


Figure 4: Model of the human body and seat

Items	Set value				
	GA	Proposed method (Analysis)			
		1	2	3	4
Tolerance of $y$		$-10 \leq y \leq 20$			
Feasible area of $t_1$		$10 \leq \theta_C \leq 25$			
Feasible area of $t_2$		$20 \leq \theta_B \leq 35$ , $\theta_B \geq \theta_C + 10$			
Feasible area of $t_3$		$0 \leq \theta_F \leq 30$			
Max iteration number $T_{\max}$	10000	100			
$c_1$	—	2.8	2.05	1.55	1.05
$c_2$	—	1.3	2.05	2.55	3.05
$w_{\min}$	—	0.4			
$w_{\max}$	—	0.9			
Number of assignable points	2	2 (number of swarms)			
Swarm size	—	20			
Solution number		5			

Table 2: Definition of parameters

## Result

The design solutions (adjustable ranges) derived by the proposed method using the different parameters and by the GA are shown in Figure 6. Additionally, the Euclidean distance between assignable points  $D$  is calculated in order to compare the size of the adjustable range. The average and the standard deviation of the distance are indicated as shown in Table 3. Figure 6 and Table 3 show the adjustable ranges derived by the proposed method are smaller than and assure as same robustness as those by the GA. Particularly, the mean value and the standard deviation of the adjustable ranges derived in analysis 2 are small. This means the values of the parameter ( $c_1 = c_2 = 2.05$ ) are suitable to minimize the adjustable range. This is caused by the two features of the VEPSO: 1) the larger  $c_2$  prevents the global solution search same as the PSO; 2) the larger  $c_2$  encourages the swarms to search the area close to each other. Because of the trade-off relationship between the two features, the same degree of incidence is compromised to be an optimum value in the proposed method.

## Conclusions

In this research, ACFs, which can be adjusted within a given range to increase the robustness, were defined. Additionally, a method to calculate the robustness  $R_A$ , including the objective characteristics adjustment by the factors and to derive an optimum range of the factors is proposed.  $R_A$  indicates the feasibility that the objective characteristic values are within the tolerance at least once by the adjustment of ACFs. A calculation methods for the index, which uses the Monte Carlo method, are proposed. In contrast, the range of ACFs is optimized by the Vector evaluated particle swarm optimization. In the procedure,  $R_A$  is used to evaluate the particles in several swarms, and each particle searches for the optimum adjustable range of ACFs.

The proposed method was applied to an engineering example (seat design problem). In this application, it was confirmed that the proposed method can derive the design solution with high robustness and small adjustable range.

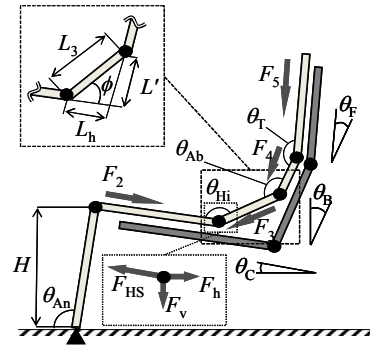
## Acknowledgement

This work was supported by the Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research (C) (23611037).

## References

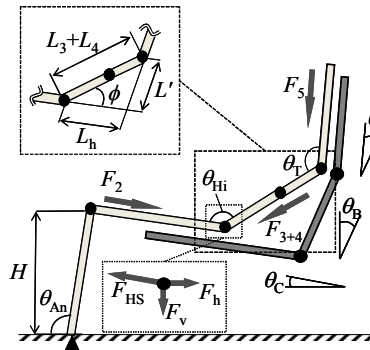
- Arakawa M and Yamakawa H (1995) A study on Optimum Design Using Fuzzy Numbers as Design Variables. In: Proceedings of the ASME DETC, DE-82: 463-470.
- Belegundu AD and Zhang S (1992) Robustness of Design Through Minimum Sensitivity. *ASME Journal of Mechanical Design* 114(2): 213-217.
- Carlisle, A. and Dozier, G. (2001). An off- the-shelf PSO, Proceedings of Workshop Particle Swarm Optimization, Indianapolis, IN,.





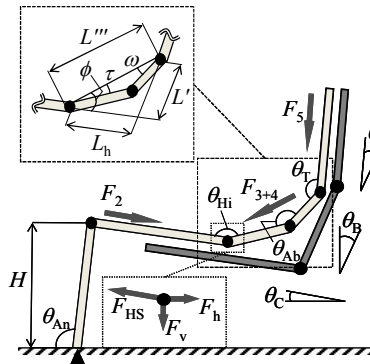
$$\begin{aligned}
 F_{HS} &= -F_h \cos \theta_C - F_v \sin \theta_C - \kappa(-F_h \sin \theta_C + F_v \cos \theta_C) \\
 \left( \begin{aligned}
 F_h &= F_2 \cos \theta_C - F_3 \sin(\theta_{Hi} + \theta_C) \\
 F_v &= F_2 \sin \theta_C + F_3 \sin(\theta_{Hi} - \theta_C) + M_2 l_{2b} g + M_3 l_{3a} g \\
 F_2 &= \frac{M_1 l_{1b} g + M_2 l_{2a} g}{\sin \theta_C - \cos \theta_C \tan \theta_{An}}, \quad F_3 = \frac{F_{4-5} + (M_4 l_{4a} g + M_3 l_{3b} g)(\cos \theta_B - \kappa \sin \theta_B)}{-\cos \theta_{Ab} - \kappa \sin \theta_{Ab}} \\
 F_4 &= (M_5 l_{5a} g + M_4 l_{4b} g)(\cos \theta_B - \kappa \sin \theta_B) + F_5(\cos \theta_F - \kappa \sin \theta_F), \\
 F_5 &= M_5 l_{5b} g \{\cos(\theta_B - \theta_F) - \kappa \sin(\theta_B - \theta_F)\} \\
 \theta_{An} &= \sin^{-1}(H/L_1), \quad \theta_{Hi} = 180^\circ - \phi, \quad \theta_{Ab} = \phi + 90^\circ - \theta_C + \theta_B - \theta_F, \\
 \phi &= \sin^{-1}\{(L'/L_3) \sin(90^\circ + \theta_B - \theta_C)\} \\
 L' &= L_h \cos(90^\circ + \theta_B - \theta_C) + \sqrt{L_3^2 - L_h^2 \sin^2(90^\circ + \theta_B - \theta_C)}
 \end{aligned} \right)
 \end{aligned}$$

(a)



$$\begin{aligned}
 F_{HS} &= -F_h \cos \theta_C - F_v \sin \theta_C - \kappa(-F_h \sin \theta_C + F_v \cos \theta_C) \\
 \left( \begin{aligned}
 F_h &= F_2 \cos \theta_C + F_{3+4} \cos(\theta_{Hi} + \theta_C), \\
 F_v &= F_2 \sin \theta_C + F_{3+4} \sin(\theta_{Hi} + \theta_C) + M_2 l_{2b} g + (M_3 + M_4) l_{ma} g \\
 F_2 &= \frac{M_1 l_{1b} g + M_2 l_{2a} g}{\sin \theta_C - \cos \theta_C \tan \theta_{An}}, \quad F_{3+4} = \frac{F_5 + (M_5 l_{5a} g + (M_3 + M_4) l_{mb} g)(\cos(\theta_B - \theta_F) - \kappa \sin(\theta_B - \theta_F))}{-\cos \theta_T + \kappa \sin \theta_T} \\
 F_5 &= M_5 l_{5b} g \{\cos(\theta_B - \theta_F) - \kappa \sin(\theta_B - \theta_F)\}, \\
 \theta_{An} &= \sin^{-1}(H/L_1), \quad \theta_{Hi} = 180^\circ - \phi, \quad \theta_T = \phi + 90^\circ - \theta_C + \theta_B - \theta_F, \\
 \phi &= \sin^{-1}\{(L'/(L_3 + L_4)) \sin(90^\circ + \theta_B - \theta_C)\} \\
 L' &= L_h \cos(90^\circ + \theta_B - \theta_C) + \sqrt{(L_3 + L_4)^2 - L_h^2 \sin^2(90^\circ + \theta_B - \theta_C)}
 \end{aligned} \right)
 \end{aligned}$$

(b)



$$\begin{aligned}
 F_{HS} &= -F_h \cos \theta_C - F_v \sin \theta_C - \kappa(-F_h \sin \theta_C + F_v \cos \theta_C) \\
 \left( \begin{aligned}
 F_h &= F_2 \cos \theta_C + F_{3+4} \cos(\theta_{Hi} + \theta_C - \tau), \\
 F_v &= F_2 \sin \theta_C + F_{3+4} \sin(\theta_{Hi} + \theta_C - \tau) + M_2 l_{2b} g + (M_3 + M_4) l_{m'a} g \\
 F_2 &= \frac{M_1 l_{1b} g + M_2 l_{2a} g}{\sin \theta_C - \cos \theta_C \tan \theta_{An}}, \quad F_{3+4} = \frac{F_5 + (M_5 l_{5a} g + (M_3 + M_4) l_{m'b} g)(\cos(\theta_B - \theta_F) - \kappa \sin(\theta_B - \theta_F))}{-\cos(\theta_T - \omega) + \kappa \sin(\theta_T - \omega)} \\
 F_5 &= M_5 l_{5b} g \{\cos(\theta_B - \theta_F) - \kappa \sin(\theta_B - \theta_F)\}, \\
 \theta_{An} &= \sin^{-1}(H/L_1), \quad \theta_{Hi} = 180^\circ - \phi + \tau, \quad \theta_T = \phi + 90^\circ - \theta_C + \theta_B - \theta_F + \omega, \\
 \phi &= \sin^{-1}\{(L'/L''') \sin(90^\circ + \theta_B - \theta_C)\} \\
 L' &= L_h \cos(90^\circ + \theta_B - \theta_C) + \sqrt{(L''')^2 - L_h^2 \sin^2(90^\circ + \theta_B - \theta_C)}, \\
 L''' &= \sqrt{L_3^2 + L_4^2 - 2L_3L_4 \cos(180^\circ - 24^\circ)}
 \end{aligned} \right)
 \end{aligned}$$

(c)

$\theta_C$ : Seat cushion angle,  $\theta_B$ : Seat back angle,  $\theta_T$ : Forward tilt angle,  $\theta_{Hi}$ : Hip angle,  $\theta_{Ab}$ : Abdomen angle,  $\theta_{An}$ : Ankle angle,  
 $\kappa$ : Coefficient of frictional resistance,  $H$ : Seat cushion height (400mm),  $F_i$ : Force on  $i$ th human body section,  
 $F_{HS}$ : Hip sliding force,  $F_h$ : Horizontal force on trochanter major,  $F_v$ : Vertical force on trochanter major,  $L$ : Body height,  
 $L_i$ : Length of  $i$ th body section,  $L_h$ : Buttock-trochanterion length,  $M$ : Body weight,  $M_i$ : Weight of  $i$ th body section,  
 $l_{ia}$ : Ratio of  $L_i$  and the length from  $i$ th body section upper-edge to gravity-center,  $l_{ib}$ :  $1 - l_{ia}$ ,  
 $l_{ma}$ : Composite ratio of 3rd and 4th body section in stretched waist sitting posture,  $l_{m'a}$ :  $l_{ma}$  in bent waist sitting posture

$$\left( \begin{aligned}
 L_1 &= 0.2880L - 0.0424, \quad L_2 = 0.0027L + 0.4057, \quad L_3 = 0.3274L - 0.2908, \quad L_4 = 0.0609L + 0.0356, \quad L_5 = 0.0930L - 0.0549, \\
 L_h &= 0.3118L - 0.4113, \quad M_1 = 0.12M, \quad M_2 = 0.2M, \quad M_3 = 0.14M, \quad M_4 = 0.18M, \quad M_5 = 0.36M \\
 l_{ia} &= 0.61, \quad l_{2a} = 0.43, \quad l_{3a} = 0.11, \quad l_{4a} = 0.11, \quad l_{5a} = 0.35, \quad l_{ma} = 0.329, \quad l_{m'a} = -0.608(L_3 + L_4) + 0.579
 \end{aligned} \right)$$

Figure 5: Hip-sliding force estimation equations

Method		$c_1, c_2$	$R_A$	$D$	$\sigma_D$
Conventional method		—	0.996	17.7	4.07
Proposed method (Analysis)	1	(2.8, 1.3)	0.998	3.92	9.66
	2	(2.05, 2.05)	0.999	0.23	0.79
	3	(1.55, 2.55)	0.999	0.49	1.41
	4	(1.05, 3.05)	0.999	1.21	4.36

Table 3 Result of analyses

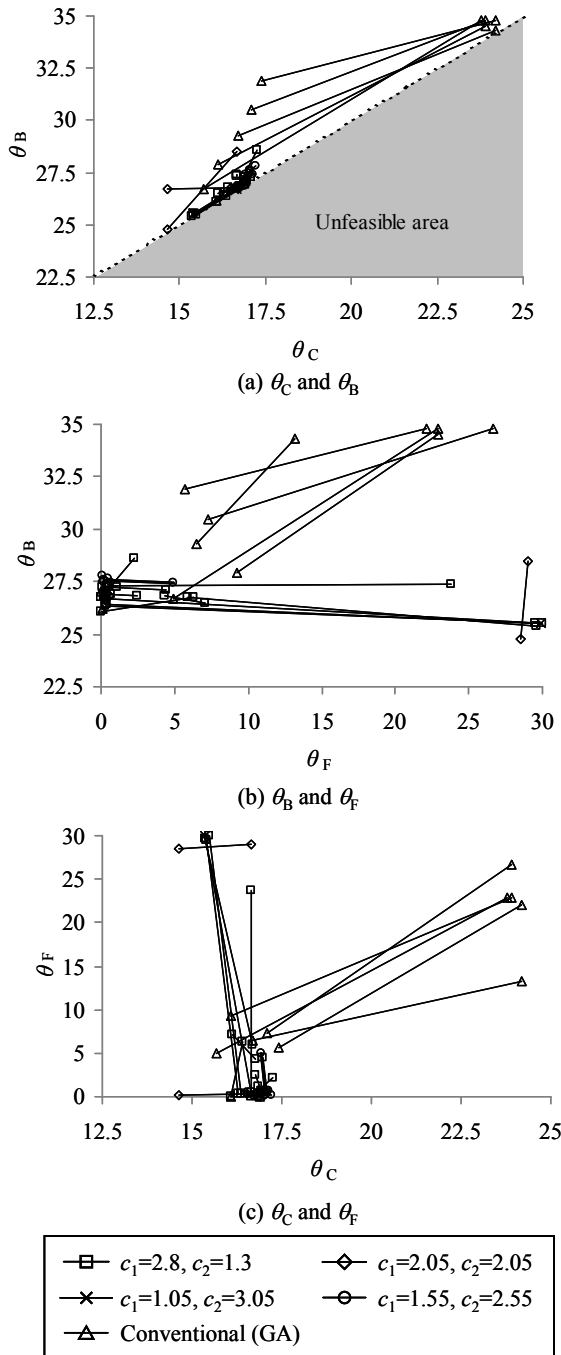


Figure 6: Solution candidates of each analysis

- Eggert RJ and Mayne RW (1993) Probabilistic optimal design using successive surrogate probability density functions. *ASME Journal of Mechanical Design* 115(3): 385-391.
- Emch G and Parkinson A (1994) Robust optimal design for worst-case tolerances. *ASME Journal of Mechanical Design* 116(4): 1019-1025.
- Kennedy, J. and Eberhart, R.C. (1995). Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks 1995, 1942-1948.
- Kennedy, J. (1998). The Behavior of Particles, Proceedings of 7th Annual Conference on Evolutionary Programming, San Diego, USA,.
- Matsuoka Y and Hanai T (1988) Study of Comfortable Sitting Posture, *SAE Technical Papers*, SAE-880054.
- Matsuoka Y (2000) Robust design method for Diversity of Ba, *KANSEI Eng Int*, 1(4): 25-32.
- Matsuoka Y (2010) *Design Science*. Maruzen, Tokyo.
- National Institute of Bioscience and Human Technology (1996) *Human Body Dimensions Data for Ergonomic Design*. Japan Publication Service, Tokyo.
- Otto KN and Antonsson EK (1993) Tuning Parameters in Engineering Design. *ASME Journal of Mechanical Design* 115(1): 14-19.
- Parkinson A, Sorensen C, and Pourhassan N (1993) A general approach for robust optimal design. *ASME Journal of Mechanical Design* 115(1): 74-80.
- Parkinson A (1995) Robust mechanical design using engineering models. *ASME Journal of Mechanical Design* 117B: 48-54.
- Ramakrishnan B and Rao SS (1996) A general loss function based optimization procedure for robust design. *Engineering optimization* 25: 255-276.
- Sundarean S, Ishii K, and Houser DR (1991) Design optimization for robustness using performance simulation programs. In Proceedings of the ASME ADA, vol DE-65-1: 249-256.
- Taguchi, G (1993) *Taguchi on robust technology development*. ASME Press, New York.
- Yu JC and Ishii K (1998a) Design optimization for robustness using quadrature factorial models. *Engineering optimization* 30: 203-225.
- Yu JC, Ishii K (1998b) Design for robustness based on manufacturing variation patterns. *ASME Journal of Mechanical Design* 120(2): 196-202.
- Vlachogiannis, J.G., and Lee, K.Y. (2005). Determining Generator Contributions to Transmission System Using Parallel Vector Evaluated Particle Swarm Optimization, *IEEE Transactions on Power Systems*, 20(4): 1765-1774.
- Vlachogiannis, J.G., and Lee, K.Y. (2009). Multi-objective based on parallel vector evaluated particle swarm optimization for optimal steady-state performance of power systems, *Expert Systems with Applications* 36: 10802.
- Watai A, Nakatsuka S, Kato T, Ujiie Y, and Matsuoka Y (2009) Robust Design Method for Diverse Conditions. In: Proceedings of the ASME DETC(DETC/CIE2009-87108).
- Zhu J and Ting KL (2001) Performance distribution analysis and robust design. *ASME Journal of Mechanical Design* 123(1): 11-17.

# The Insect Mushroom Bodies: a Paradigm of Neural Reuse

Paolo Arena<sup>1</sup>, Luca Patané<sup>1</sup> and Roland Strauss<sup>2</sup>

<sup>1</sup>DIEEI - University of Catania, Italy

<sup>2</sup>Dept. of Zoology III University of Mainz, Germany

parena@dieei.unict.it, lpatane@dieei.unict.it, rstrauss@uni-mainz.de

## Abstract

This paper is devoted to discuss the implementation of models, which are inspired by the fly *Drosophila melanogaster* and able to handle open problems in the field of robotics such as attention, expectation and sequence learning. The role of the Mushroom Bodies (MBs) in solving these tasks is analyzed in detail and a unifying plausible biologically inspired model is proposed. The developed neural structure is able to show different capabilities in line with the paradigm of neural reuse. The same neural circuit can be exploited to accomplish multiple tasks showing interesting capabilities such as attention, expectation and delayed match-to-sample. The simulation results here reported suggest at the same time new neurobiological experiments suited to better understand the underlying mechanisms, to verify the hypotheses formulated and to prove the biological significance of the results.

## Introduction

Efforts to build efficient and adaptive machines stimulated a lot of researchers to take inspiration from Nature for designing, modeling and implementing bio-mimicking circuits and systems able to reproduce specific biological behaviors: locomotion, learning, recognition and others. Since the beginning of this scientific wave, which dates back to the early part of the last Century, two main approaches were established. The first one belongs to the field of artificial intelligence; it took the high level capabilities of living beings as starting point and aimed at designing abstract, yet sometimes very well performing computational models. The second one, called connectionism, started from modeling the structure of the brain at different levels of resolution, with the claim that a good model of the low-level topology and function should lead to the emergence of behaviors well mimicking the biological counterpart, even with regard to high-level functions (Rumelhart and McClelland, 1986). Very recently this second research field successfully exploited innovative tools and methodologies from neurobiology and neurogenetics, with an effectiveness impossible to predict only two decades ago. These new tools opened the way to novel insight into the brain and tremendously contributed to unravel a lot of surprising functions of neural tissues. One of these findings is the so-called *neural*

*reuse*. The term refers to a very common property of neural assemblies, i.e. neural circuits established for one purpose are exploited, recycled, redeployed, during evolution or individual development for other different purposes, often without losing their original functions (Anderson, 2010). This research field is widely supported both by physiological observation and by imaging experiments (i.e. fMRI), where in a considerable number of cases especially high-level cognitive functions are involving the concurrent activation of different areas of the brain otherwise (and well known to be) involved in completely different functions. Insects possess a much simpler brain structure than mammals: their brains were miniaturized during evolution in response to constraints like energy consumption. Notwithstanding their much scaled-down brain size, insects are able to show an impressive number of adaptive behaviors, until a few years ago ascribed only to higher animals (Chittka L., 2009): the smaller brain size did not prevent important capabilities. Considering the fruit fly *Drosophila melanogaster*, one of its first-studied forms of learning is related to olfaction. This learning process has been localized in the Mushroom Bodies (MBs), one of the two prominent insect brain neuropiles of the protocerebrum; the other one is the Central Complex (CX). MBs are commonly considered as a model system for the biochemistry and the connectivity of how synaptic networks can form memory and store information. MBs are responsible for both the short-term and long-term component of olfactory memory. Several different experiments demonstrated how, through classical conditioning, flies can associate a meaning to olfactory inputs after pairing them with positive and negative reinforcement signals (Gerber et al., 2004). Compared to the rest of the insect brain, MBs have attracted a lot of attention witnessed by the recent specialized literature, since, besides olfactory processing and learning, recent studies identified MBs as also responsible for other learning processes (Scherer et al., 2003; Liu, 2006) and choice behaviors (Gronenberg and Lopez-Riquelme, 2004; Tang and Guo, 2001; Brembs, 2009). By modeling the MBs as a pool of spiking neurons divided into different lobe systems and introducing the synaptic connec-

tions identified between the MB intrinsic Kenyon cells and the other structures directly involved like Projection Neurons (PN), Antennal Lobes (AL) and the Lateral Horn (LH), it is possible to investigate the emergence of interesting neural activities that can establish specific behaviors shown in flies, like attention (Arena et al., 2012b), expectation (Arena et al., 2012a) and delayed-matching to sample tasks (Arena et al., 2013) that have been recently proposed. The role of MBs in motor learning is also known from fly experiments: in particular the short-term memory component is not obtained if MB plasticity is inhibited in mutant flies. MBs can be modeled as a reward-driven parameter adapter that improves the fly performance while a task like climbing over a chasm is continuously repeated for multiple times (Pick and Strauss, 2005).

This paper revisits a recently introduced low-level model of the fruit fly *Drosophila melanogaster* MBs under the aspect of neural reuse. It will be shown how the same neural structure can concurrently give rise to a number of different adaptive behaviors, which are also encountered in the biological counterpart. Neural reuse in action will be shown referring to behaviors ranging from classical conditioning, to attention, expectation, consolidation and delayed-matching-to-sample. All these last capabilities can be easily transferred to robotic structures for the implementation of real-time adaptive behaviors.

### Neural Reuse Theory

Various general theories were proposed on the overall functionality of the brain. Among them are two main lines of research worth to be considered in the context of this paper: *Massive Modularity* and *Neural Reuse*. The first theory is mainly drawn from Evolutionary Psychology (Sperber, 2001) and claims that brain processing can be studied by decomposing it into dissociable functional components that vary independently of one another (Carruthers, 2006). Indeed this is a very useful approach, especially when it comes to dealing with complex brains: decomposing and localizing modular blocks in some cases allowed focalizing on specific brain functions. On the other hand, the Neural Reuse theory of brain processing appears to be radically different. It starts from the fact that brains are complex dynamical systems. It refrains from the idea to functionally break a complex function in sub-functions and to assign these functions to specific parts, but rather applies a holistic and somehow heuristic approach to brain functions, which has a lot in common with the complex dynamical system theory. It is grounded on the concepts of *network thinking* and pays attention to higher order brain functions as patterns of neural activity emerging from the overall behavior of a complex system, caused by the spatio-temporal, self-organised, synchronized activity of different parts of the brain working as an orchestra. This concept has been recently studied from the physical point of view especially in simple brains, like the one of the worm

*C. elegans*. These studies demonstrated that network theory can topographically and dynamically address brain dynamics, at least in those small neural assemblies (Dunn et al., 2004). According to the complex system approach, brain behavior can be described in the language of Patterns, and it is most powerful when several, particularly higher brain areas are involved. The decomposition approach is powerful when specific, mostly sensory-input related processing can be broken up into specialized functions. On the other hand decomposition can be too restrictive when looking for much higher functions, like learning, decision making, multisensory processing and complex sensory-motor loops. Within this perspective of complex tasks, if multiple brain areas are involved, it directly derives that various behavioral purposes have to be achieved concurrently. This means that the same spatial temporal patterns emerging from one neural lattice are exploited at the same time in multiple behaviors. This is the core of the *Neural Reuse Theory* (Anderson, 2010). This concept appears to be radically different from the massive modularity concept, even if also neural reuse accepts some functional bias within individual brain regions, especially for those dealing with specific sensory features; it poses a specific distinction between the concept of *work* and that of *use* (Bergeron, 2008). The former is related to the fixed low-level functions of specific brain regions whereas the latter refers to the way these workings are arranged together for many different and concurrent uses of those same regions. The concept of neural reuse can be considered in a phylogenetical and ontogenetical perspective: phylogenetically driven, in the sense that evolutionary processes are biased toward using already functioning circuits over introducing new ones; ontogenetically driven, since learning, with the addition of neural wiring, is one powerful possibility to connect different brain regions and creates cross-over associations. Along these lines, the *Massive Redeployment Hypothesis* (Anderson, 2007) assesses neural lattices that are massively re-used in different high-level functions, since they can be connected in different ways, leading to very different functions. We'll try to apply the introduced neural reuse paradigm on a specific example, a multifunctional structure of the insect brain called Mushroom Bodies. Citti and Niven (Niven and Chittka, 2010) already asserted that insect brains have the suitable size to try to investigate neural reuse in action, since in these small brains, the relatively low number of neurons and mainly short-distance connections are candidate elements for neural reuse, even if circuits are composed of different brain areas. For example, a retention of aversive olfactory memory from larvae to adult flies (Tully et al., 1994) suggests a reuse of particular neural structures through metamorphosis. The smaller the brain, the larger the need for neural reuse. Even if anatomical modularity is clearly given in such small brains, it is also true that a lot of sensory-motor behaviors involve different brain sectors. The insect brain thus appears to be a net-



work mainly composed of locally connected circuits, which however are connected through rare, even single-unit made, long-range links. Whereas modularity increases energy efficiency, the presence of long-range connections appears to promote neural reuse. Therefore invertebrates are a suitable class of animals for neural-reuse investigation, and also for finding an expected compromise between massive modularity and reuse.

### Mushroom Bodies: Neurobiological aspects

The MBs of the fruit fly *Drosophila melanogaster* MBs are a paired structure of the protocerebral hemispheres. The most important constituents of the MBs are the 2500 Kenyon cells (KCs) per side which run in parallel from the input-region calyx through the peduncle and, after a bifurcation, to specific appendices, called lobes. These possess roughly the same topology, but are differently connected to the other neural structures. Among them are the  $\alpha - \beta$ - and the  $\alpha' - \beta'$ -lobes. In flies, there is a prominent olfactory input from the antennal lobes into the calices (Masse et al., 2009). Input from other sensory modalities is not topologically identified in *Drosophila*, but the role of MBs in tasks related to vision, decision making and behavioral adaptation have been reported e.g. in Liu et al. (1999) and Tang and Guo (2001). In the insect brain, MBs interact with Lateral Horns (LHs) and Antennal Lobes (ALs). Recent studies have shown that mutations affecting olfactory-memory formation in *Drosophila* also produce distinct defects in visual attention-like behaviors (van Swinderen and Flores, 2007), suggesting that parts of MBs are reused in several different behavioral contexts and across several sensory modalities. MBs and LHs codify the spatio-temporal information coming from the glomeruli of the ALs. Connections between LHs and MBs have been found, whose entity in *Drosophila* is not well known, but in locusts, which produce an inhibitory effect to the MBs neurons (Perez-Orive et al., 2002); in *Drosophila* they are not yet identified. Not anatomically obvious in *Drosophila*, but in honeybees, MBs receive inputs from other sensory modalities but olfaction like vision, gustation and mechanosensation. In flies and bees, the MB lobe region receives information on sugar reward or electric shock through octopaminergic and dopaminergic neurons, respectively. There is an output of the MBs to pre-motor areas of the brain.

A general block scheme of the interactions among the different neural structures involved in the proposed model is depicted in Fig. 1.

Inside the MBs the flow of information is through the Kenyon cells from the calyx towards the lobes. Neuroanatomical studies in *Drosophila* revealed arborizations of extrinsic and intrinsic MB neurons across the peduncle and mainly in the lobe systems. The lobes are the output region of the MBs and also a region for modulatory inputs (Krashes et al., 2009). Intrinsic neurons provide an alterna-

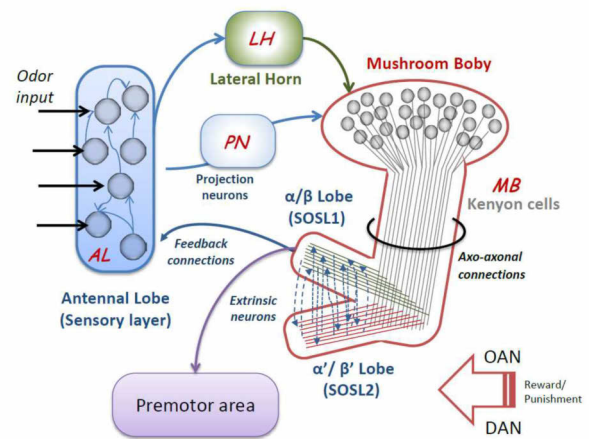


Figure 1: MBs and their interactions with other insect brain centers. MBs, together with the antennal lobes (ALs) and the lateral horn (LH), are the place for odor representation and learning. The presence of axo-axonal connections between the Kenyon cells, feedback connections between the lobes and the AL layer and the reward or punishment signals mediated via octopaminergic (OAN) and dopaminergic (DAN) neurons, respectively, are important elements that allows the emergence of patterns of neural activity responsible for multiple complex behaviours.

tive modulation pathway between different KCs and/or KCs and other protocerebral brain areas. Extrinsic neurons, on the other side, may be able to bind sensory information processed earlier in different lobes before or after any kind of modulation. This is very interesting for our modeling purposes. Very recently, recurrent connections between MBs and ALs have been found (Hu et al., 2010). The presence of this functional feedback from the MBs to the ALs suggests top-down modulation of olfactory information processing in *Drosophila*. The presence of dynamically changing conditions and noise in the environment leads animal to develop attention-like processes. Attention facilitates focusing on the attended events, while filtering out irrelevant information. These interesting processes have been studied in *Drosophila*. In particular, results by Xi et al. (2008) suggest that MBs in flies behave like an adaptive sensory gain controller, allowing the processing of salient cues, filtering out the background noise and distracting signals. More information from neurobiology, essential to develop an efficient, flexible and multi-functional neural model, concerns the presence of axo-axonal connections among the kenyon fibres, whose role could not be clearly unraveled in experiments, but give us the possibility to add to the computational model efficient diffusion phenomena, which are at the basis of the spatial-temporal clustering capabilities.

### Mushroom Bodies: a computational model

The proposed computational model is directly inspired by the MB structure, including top-down connections to the Antennal lobes, the global inhibitory effect of the Lateral Horn and the axo-axonal diffusive connections among the KC fibers. The proposed neural architecture is a two layers recurrent network in which each neuron is an Izhikevich's class one spiking neuron with spike resetting, (Izhikevich, 2003), which offers many advantages over other neuron models from the computational point of view. Neurons are connected through synapses, modeled as first-order dynamical systems which transform the pre-synaptic voltage spike trains into a post-synaptic current. There are sites where learning is added to the basic dynamics of the synapses. These adaptive sites are areas where neural reuse is prominent: reuse exploits the dynamics arising at sites (which correspond to the working sites in (Anderson, 2010)) where learning is not present, but synaptic and neural dynamics concur to the emergence of clusters of neural activity. Learning is implemented through a correlation based, Hebbian rule called Spike Timing Dependent Plasticity (STDP) (S. Song, 2000, 2001) which has been used in different application, including robot learning (Arena et al., 2009). The algorithm acts on the synaptic weights, modifying them according to the temporal sequence of occurring spikes (Arena et al., 2011). An output layer can be added in order to link the behavior of the second layer to a motor or pre-motor area. The developed neural structure, even if inspired by the insect olfactory system, can be used for stimuli of different sensory modalities (e.g visual features can be easily used in robotic scenarios).

The Antennal Lobes model, as shown in Fig. 1, receives input from the olfactory sensory system (i.e. Antennas). We can assume that each AL neuron, when active, codifies the presence of a particular value for a specific feature of the input. Neurons in the ALs are organized in groups, each group presenting the different values of a given input feature. A competitive topology is implemented between neurons within each group, whereas plastic excitatory synapses link neurons belonging to different groups. When the AL layer is stimulated, after a short transient time, only one neuron in each group remains excited, owing to a Winner Takes All (WTA) topology. The ensemble of all the active neurons encodes the presented object. The adaptive connections between groups of neurons bias the network toward temporarily retaining the presented objects through all its features or to reconstruct lacking features in case of incomplete or noisy detection. Non-learning synapses from the AL model to the MB model are randomly established, with a given probability of connectivity. The MB model is made-up of two twin lattices representing the two-lobes system, here called Self Organizing Spiking Layer (i.e. *SOSL1* and *SOSL2*). They have a toroidal geometry, with local excitatory and global inhibitory synapses. Each neuron within them is connected

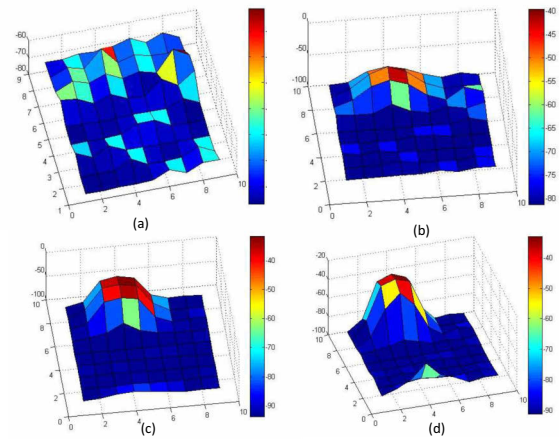


Figure 2: Time evolution of the neural activity in the Self Organizing Spiking Layer (SOSL) lattice. The network topology allows the emergence of a winning cluster. This example is related to a lattice of 9x9 neurons (represented in the x-y plane) and reports the mean potential of each neuron in the lattice (z axis) evaluated by dividing the simulation in four time windows. The clustering is obtained after 80 ms of simulation.

through fast excitatory synapses with all the neurons of its neighborhood and through fast inhibitory synapses with all the other neurons of the lattice. The main peculiarity of these SOSLs is a spontaneous clustering, due to the competitive topology: information coming from the ALs is compressed into a cluster of spiking activity, which can arise in different positions in the SOSL1 and SOSL2 due to the random connectivity of the SOSLs with the input layer. A typical neural activity leading to the emergence of a cluster in the SOSL is shown in Fig. 2.

Only in the SOSL1, representing the  $\alpha - \beta$  lobes, a slow and delayed diffusion of the neural activity links each neuron to the other neurons of the same lattice, in order to have the possibility to temporally link different clusters. These synapses are subjected to the STDP learning algorithm, that allows discovering and retaining temporal causality among clusters. These connections have the interesting capability to generate, within the SOSL1 layer, expectation and short-term prediction capabilities, whereas the SOSL2 layer has the main function of working as a back-up copy. Feedback connections able to learn link clusters in SOSL1 to the ALs neurons.

The two lobes (SOSLs) are connected to each other through two sets of plastic synapses, one from the  $\alpha - \beta$  lobes to the  $\alpha' - \beta'$  lobes and the other set from the  $\alpha' - \beta'$  to the  $\alpha - \beta$  lobes. It is known from neurobiology that the neurons belonging to the two clusters are morphologically different. Moreover, whereas  $\alpha - \beta$  neu-

rons give information back to the ALs, by that generating a feedback at the sensory stage, the  $\alpha' - / \beta'$  neurons were found to provide no output signals back into the system, but they just receive sensory input in the last place at the level of the calices. Arena et al. (2013) therefore assumed that the information which arrives at these lobes is retained there and used as a kind of backup copy for memory purposes. Further details on the model are reported in (Arena et al., 2012a).

To summarize this paragraph, input coming from the ALs is clustered into feature-based objects. This information is sent, without learning, to the SOSLs where a clustering effect arises in terms of self-organized spiking activity. The SOSLs concurrently integrate the information coming from the ALs in a given time window: in particular after the emergence of a cluster the neural activity of the network is inhibited by the LH wave. Clusters of spiking in SOSL1 are temporally linked, thanks to synapses that are able to learn, and concurrently, a cluster-induced depolarization of the AL layer is emerging like in Fig. 3. Moreover, a cluster corresponding to the previous input provides information on the past experience of the network for sequence learning purposes. One crucial issue is the synchronization between the delay of the synaptic cluster-linking connections in the SOSLs and the LHs induced inhibitory action onset time. The output layer connects the model to a motor or premotor area. Neurons in the output layer are linked to the SOSL neurons through synapses subject to an associative learning.

### Behavioral repertoire

The concept of neural reuse is applied to the MB architecture that is able to show different behaviors in a unique dynamical system.

### Conditioning in MBs

The model works for conditioning purposes exploiting the feedforward processing of the network. Clusters emerging in the SOSL1 after presenting a given input stimulate an output associative output neuron. When a rewarding unconditioned stimulus is given to the network, a Reward Neuron (RN) becomes active. This is connected to the output neuron through a fixed synapse, representing the unconditioned response to the reward. That way the output neuron is forced to fire, and the synapse connecting this neuron to the SOSL1 cluster is trained. The mechanism guides classical and operant conditioning through hebbian learning: the output neuron takes the role of a premotor neuron, in case of applications to tactic or phobic reactions following learned attractive or repulsive signals, respectively, in robotic applications.

### Modeling attention using the Mushroom Body structure

The attention capability was well assessed experimentally by van Swinderen (2011). Attentional capabilities are ob-

tained by exploiting the feedback synapses, which reuse the dynamics formed into the  $\alpha - / \beta$  lobes lattice (SOSL1) to provide an input to the AL layer. The role of feedback connections in the insect olfactory system model has been analyzed by Arena et al. (2011), on the basis of the biological evidences found by Hu et al. (2010). Feedback synapses are updated according to the STDP learning algorithm. When a cluster is elicited in the SOSL1, the synaptic connections between these neurons and those neurons which are firing in the AL (due to the synchronous presence of the corresponding input) are strengthened, according to the Hebbian paradigm. This produces a pre-polarization of the AL layer and acts as a filter for the sensory input, leading to an attention-like phenomenon. Two major aspects are worth mentioning: the first one is that the spatio-temporal dynamics assumed to link the different internal representations of objects in time is modeled as a specific function of the SOSL1 lobe. The second aspect is that the output of the feedback synapses is delayed; the postsynaptic current influences the input layer only after the lobes have been reset via the LHs. This is equivalent to assume that the action of these feedback connections, being focused to enhance attention loops, is able to persist also after the inhibition coming from the LHs. The actual model hypothesizes massive feedback connections from SOSL to ALs neurons, even if a probability distribution could maintain the same performance in case of a large-scale implementation.

### Modeling expectation

During the efforts spent in delivering an MB model able to elicit both traditional odor conditioning features and attention processes, a behavior emerged, which is related to our concept of expectation. Indeed, this capability is not yet found in insects, but nevertheless the computational results can open the way to a new wave of insect experiments in this direction. The MB neural lattice (SOSL1) is reused for creating expectation by means of the set of plastic delayed synapses (providing a kind of second order diffusion effect) linking each neuron of the SOSL1 to the other neurons of the same lattice, in order to have the possibility to temporally link different clusters. These synapses are subject to the STDP learning algorithm, that allows discovering and retaining temporal causality among clusters. Simulation results show that these connections have the interesting capability to generate expectation and short-term prediction features within the SOSL. The plastic feedback connections, present from the SOSL1 to the AL model and exploited for attention, have here a precious role (this can be also considered a kind of reuse) to boost the model performance. Two main functions have been identified: they are useful to create an expectation-based depolarization of the neurons in the ALs and they are also essential to reconstruct the expected object.

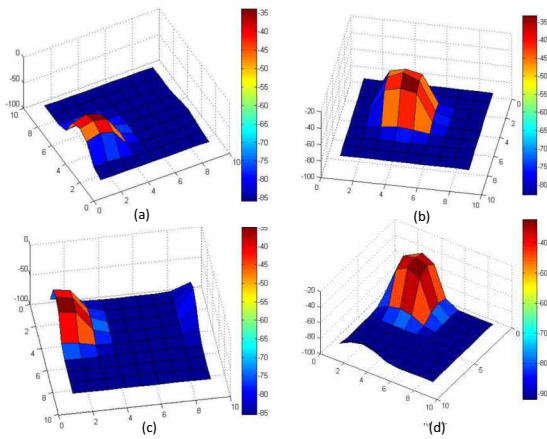


Figure 3: Time sequence of evoked winning clusters obtained during memory consolidation. Each plot represents the mean potential of the voltage in a 9x9 lattice in a steady state condition. The role of noise is important to give to the lattice enough energy to allow the rise of a cluster (a) that activates in time the different clusters (b-d) in a previously learned sequence; this consolidates the memory.

### Memory consolidation

By exploiting the presence of noise in the system (Arena et al., 2012a), an interesting property of the network can emerge. In fact, the contribution of noise can be useful to consolidate the acquired knowledge during a resting phase, when the network is not physically connected to any input signal. In this phase that can be thought of as sleeping condition, the network is simulated no longer by real-world signals used for training, but by noise. The simulation aimed at showing this effect consists first of a learning phase, in which the network creates the association between clusters and objects. As discussed in (Arena et al., 2011) at the end of the training phase no physical input is presented at the ALs layer, but it is assumed that the SOSL1 is subject to noise. These disturbances onset transients in the SOSL1 neurons until a cluster will emerge over the others. If this cluster had formerly been trained to represent a given object, this is recalled at the AL layer, like an “evoked object”. An example of sequence of evoked clusters is reported in Fig. 3.

A new learning cycle will then arise, in which not only this object will be consolidated, but also all the other objects expected after this one in an already learned sequence. During this simulation phase, the network is reused: new imagined solutions could also be experienced during this “sleeping phase”; the system could even create new or longer sequences starting from what it already learned during the “awake phase”.

### The delayed matching-to-sample task

The previously introduced model has been extended to include the role of the MB-lobes in solving problems like the delayed matching-to-sample task: the capability to recognize successive presentations of the same object (Arena et al., 2013). For the first time now this behavior directly exploits the presence of two different lobes structures:  $\alpha - \beta$ — and  $\alpha' - \beta'$ —lobes. It is known from neurobiology that the neurons belonging to the two clusters are morphologically different: whereas  $\alpha - \beta$ — neurons give information back to the ALs, by that generating a feedback to the sensory stage, the  $\alpha' - \beta'$ — neurons were found to provide no output signals back into the system, but they just receive sensory input at the level of the calices. It was therefore assumed by Arena et al. (2012a) that the information arriving at these lobes is retained there and used as a kind of backup copy for memory purposes. As presented in the previous section, each lobe is modeled as a toroidal lattice with clustering capabilities. In addition to the delayed feedback synapses from the  $\alpha - \beta$ — lobes to the input layer, and useful for attention capabilities, the lobes are connected to each other through two sets of plastic synapses, one from the  $\alpha - \beta$ — lobes to the  $\alpha' - \beta'$ — lobes and the other set from the  $\alpha' - \beta'$ — to the  $\alpha - \beta$ — lobes. The overall process dynamics develops as follows. Each SOSL network shows a cooperative-competitive dynamics: if excited, the neurons in both SOSL networks begin a competition and, after a transient, only one cluster of neurons will remain active and stable in each lobe. The LH inhibits both networks after every time window. The resetting effects only the soma, but not the spike responses of the synapses; in particular, those ones between the lobes are reinforced when two clusters in different lobes are concurrently active. This creates a positive loop which increases the spiking rate in the SOSL1 active cluster. We assumed also to have a neuron sensitive to the firing activity of the  $\alpha - \beta$ — lobes network to detect this situation. This structure was used in (Arena et al., 2012a) to detect whether the object presented to the input layer remains the same in two subsequent steps. In fact, the successive presentation of a different object does not cause the closing of the loop and therefore prevents any increase of the synaptic activity in the lobes.

### Towards other behaviors

Insect MBs are involved in a lot of different behaviors: we are in the process of integrating most of these behaviors into a unique model. First of all sequence learning, a capacity of bees, can be considered as an iterated form of the expectation process. Moreover, the concept of sameness, found also in the behavioral repertoire of bees, can be seen as an augmented form of the delayed matching-to-sample-task: we are not so far from showing also these features within the proposed architecture. Even more complex behaviors, within which MBs have a clear role as in decision making



(Tang and Guo, 2001) and motor learning, are presently being deeply investigated. We already have working models that can explain each one of these capacities, but our aim is to use as much as possible information from neurobiology to devise a unified model for all of them, as it is done in the biological counterpart.

## Conclusions

Insects show a complex behavioral repertoire and, in recent years, are becoming a reference point in neuroscience. Their tiny brains must serve all the survival operations despite their really small mass: the surprise is that also a lot of behaviors that are summarized briefly in this paper and traditionally ascribed to the brains of higher animals, are experimentally found in insects as well. Mushroom bodies, the most studied neural assemblies in the insect brain, have several times been functionally compared to mammalian brain centers like the hippocampus (because of their involvement in learning and memory) and the cerebellum (for their involvement in motor learning). MBs play an important role in a large number of behavioral capacities and it is apparent that they serve different low- and high-level functions concurrently: therefore, in accord with the theory of neural reuse, MBs are a paradigmatic case of reused neural networks in action. The role of extrinsic neurons appears to be fundamental, some of which have connections to the KC fibers at the level of the MB calices and others at the level of the lobes and at the same time to other brain centers like premotor areas, ALs and LHs. They appear as the natural candidates to exploit the neural dynamics within the MB cells to boost the insect brain functionality.

In this manuscript we presented a model that has been recently introduced, but has been revisited here in terms of multi-functionality and neural reuse. This model was built in a connectionist manner, obeying, although in a scaled version, the neurobiological topology. The model was initially built for showing basic learning and conditioning capabilities; subsequently it was found able to show other interesting behaviors, like attention, expectation, sequence learning, consolidation during sleep and delayed-matching-to sample tasks. All of the just-mentioned features emerge from the same spiking neural lattice, which is reused in different ways. An interesting fact is that, while most of these behaviors were experimentally found in fly experiments, others, like expectation, were not yet considered as a capability of the fly. The fact that the model built here is able to show such additional capacities, opens the way to design experiments for finding these behaviors in flies. This is a win-win example in which neurobiology and computational modeling can mutually help one another to advance knowledge in both fields. There are other complex behaviors that these tiny insect brains are able to show, like decision making, motor learning and so on, in which MBs are known to be involved. Particular efforts are ongoing to see if and to

what extent the actual model might be able to represent also these additional behaviors; these efforts will further enhance our understanding of the concept of neural reuse and multifunctionality.

## Acknowledgements

This work was supported by EU Project EMICAB, grant no. 270182.

## References

- Anderson, M. (2007). The massive redeployment hypothesis and the functional topography of the brain. *Philosophical Psychology*, 21:143–174.
- Anderson, M. (2010). Neural reuse: A fundamental organizational principle of the brain. *Behavioral and Brain Sciences*, 33:245–313.
- Arena, P., Fortuna, L., Frasca, M., and Patané, L. (2009). Learning anticipation via spiking networks: application to navigation control. *IEEE Transaction on Neural Networks*, 20(2):202–216.
- Arena, P., Patané, L., Stornanti, V., Termini, P., Zaepf, B., and Strauss, R. (2013). Modelling the insect mushroom bodies: Application to a delayed match-to-sample task. *Neural Networks (special issue on Autonomous Learning)*, 41:202–211.
- Arena, P., Patané, L., and Termini, P. (2012a). Learning expectation in insects: a recurrent spiking neural model for spatio-temporal representation. *Neural Networks*, 32:35–45.
- Arena, P., Patané, L., and Termini, P. S. (2011). An insect brain inspired neural model for object representation and expectation. In *International Joint Conference on Neural Networks (IJCNN 2011)*, San Jose California.
- Arena, P., Patané, L., and Termini, P. S. (2012b). Modeling attentional loop in the insect mushroom bodies. In *International Joint Conference on Neural Networks (IJCNN 2012)*, pages 7–12, Brisbane, Australia.
- Bergeron, V. (2008). Cognitive architecture and the brain: Beyond domain-specific. *Unpublished doctoral dissertation*, Available at: <http://circle.ubc.ca/handle/2429/2711>.
- Brembs, B. (2009). Mushroom bodies regulate habit formation in *Drosophila*. *Current Biology*, 19(16):1351–1355.
- Carruthers, P. (2006). *The architecture of the mind: Massive modularity and the flexibility of thought*. Clarendon Press/Oxford University Press.
- Chittka L., N. J. (2009). Are bigger brains better? *Current Biology*, 19:R995–1008.
- Dunn, N. A., Lockery, S., Pierce-Shimomura, J., and Conery, J. (2004). A neural network model of chemotaxis predicts functions of synaptic connections in the nematode *Caenorhabditis elegans*. *Journal of Computational Neuroscience*, 17:137–147.
- Gerber, B., Tanimoto, H., and Heisenberg, M. (2004). An engraving found? Evaluating the evidence from fruit flies. *Current Opinion in Neurobiology*, 14(6):737–744.

- Gronenberg, W. and Lopez-Riquelme, G. (2004). Multisensory convergence in the mushroom bodies of ants and bees. *Acta Biologica Hungarica*, 55:31–37.
- Hu, A., Zhang, W., and Wang, Z. (2010). Functional feedback from mushroom bodies to antennal lobes in the *Drosophila* olfactory pathway. *Proceeding of the National Academy of Science of the USA*, 107(22):10262–10267.
- Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 14(6):1569–1572.
- Krashes, M., DasGupta, S., Vreede, A., White, B., Armstrong, J., and Waddell, S. (2009). A neural circuit mechanism integrating motivational state with memory expression in *Drosophila*. *Cell*, pages 416–427.
- Liu, L., Wolf, R. Ernst, R., and Heisenberg, M. (1999). Context generalization in *Drosophila* visual learning requires the mushroom bodies. *Nature*, 400:753–756.
- Liu, X.; Davis, R. (2006). Insect olfactory memory in time and space. *Current Opinion in Neurobiology*, 6:679–685.
- Masse, N., Turner, G., and Jefferis, G. (2009). Olfactory information processing in *Drosophila*. *Current Biology*, 19:700–713.
- Niven, J. and Chittka, L. (2010). Reuse of identified neurons in multiple neural circuits. *Behavioral and Brain Science*, 33:285.
- Perez-Orive, J., Mazor, O., Turner, G., Cassenaer, S., Wilson, R., and Laurent, G. (2002). Oscillations and sparsening of odor representations in the mushroom body. *Science*, 297:359–365.
- Pick, S. and Strauss, R. (2005). Goal-driven behavioral adaptations in gap-climbing drosophila. *Current Biology*, 15:1473–8.
- Rumelhart, D. and McClelland, J. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*. Cambridge, MA: MIT Press.
- S. Song, L. A. (2001). Cortical development and remapping through spike timing-dependent plasticity. *Neuron*, 32:339–350.
- S. Song, K.D. Miller, L. A. (2000). Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3:919–926.
- Scherer, S., Stocker, R., and Gerber, B. (2003). Olfactory learning in individually assayed *Drosophila* larvae. *Learning and Memory*, 10:217–225.
- Sperber, D. (2001). *In defense of massive modularity*. Cambridge, MA: MIT Press.
- Tang, S. and Guo, A. (2001). Choice behavior of *Drosophila* facing contradictory visual cues. *Science*, 294:1543–1547.
- Tully, T., Cambiazo, V., and V., K. (1994). Memory through metamorphosis in normal and mutant *Drosophila*. *Journal of Neuroscience*, 14:68–74.
- van Swinderen, B. (2011). Attention in *Drosophila*. *International Review of Neurobiology*, 99:51–85.
- van Swinderen, B. and Flores, K. (2007). Attention-like processes underlying optomotor performance in a *Drosophila* choice maze. *Developmental Neurobiology*, 67:129–145.
- Xi, W., Peng, Y., Guo, J., Ye, Y., Zhang, K., Yu, F., and A., G. (2008). Mushroom bodies modulate salience-based selective fixation behavior in *Drosophila*. *European Journal of Neuroscience*, 27:1441–1451.

# Ballistic Action Planning in Robotics by means of Artificial Imagery

Alessandro Di Nuovo<sup>1,2</sup>, Davide Marocco<sup>1</sup> and Santo Di Nuovo<sup>3</sup> and Angelo Cangelosi<sup>1</sup>

<sup>1</sup>Plymouth University, United Kingdom ; <sup>2</sup>Università di Enna "Kore", Italy ; <sup>3</sup>Università di Catania, Italy

## Abstract

### Introduction

This abstract aims to present our recent work on exploring the concept of mental imagery and mental simulation as a fundamental cognitive capability applied to robot controllers, with the aim of improving the motor performance of the robot in terms of motor control and multi-degrees of freedom coordination. Indeed we believe that mental imagery models can give the opportunity to apply such behaviour toward the development of artificial cognitive systems, in order to improve robots' motor performance in general and in complex motor planning. This objective can be achieved using bio-inspired computational modelling technologies, such as artificial recurrent neural networks, able to emulate processes of mental training by mental simulation.

In particular, as proof-of-concept, we designed a dual neural network architecture, that allows the iCub to improve autonomously its sensorimotor skills, with techniques inspired by the ones that are employed with human subjects in sports training. This is achieved by endowing a feedforward controller of a secondary recurrent neural system that, by exploiting the sensorimotor skills already acquired by the robot, is able to generate additional *imaginary* examples that can be used by the controller itself to improve the performance through a additional learning process. Moreover we show that data obtained with artificial *imagination* could be used to simulate *mental training* to learn new tasks and enhance their performance. Results of experimental tests in controlling a ballistic movement with the simulator of the iCub humanoid robot platform are presented as evidence of the opportunities presented by the use of artificial mental imagery in cognitive robotics.

### Material and Methods

The neural system that controls the robot is represented in Figure 1(a), that consists of a three layer feedforward network (FFNN) that implements the actual motor controller, and of a Recurrent Neural Network (RNN). The RNN models the motor imagery and it is represented in detail in Figure 1(b). Normalized joint position of shoulder pitch, torso

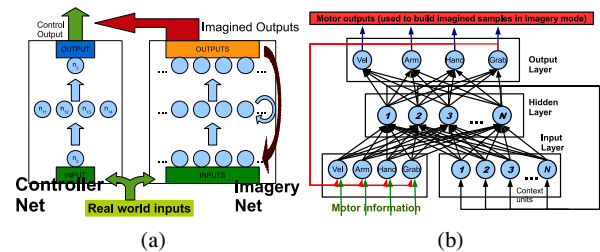


Figure 1: Artificial Neural Networks: (a) The Dual Network Architecture (FFNN + RNN). (b) Detail of RNN: Red connections are active in *imagery mode* only, while green connections are deactivated in *imagery*.

yaw, and hand wrist pitch are the proprioceptive information for input and output neurons. Another neuron is the grab/release command, respectively with value 1 and 0. All values are normalized in the range [0,1]. We used a classic back-propagation algorithm as the learning process. The learning phase lasted 10000 epochs with a learning rate of 0.2 without momentum. The experimental task is shown in Figure 2(a) and it is the realisation of a ballistic action, involving the simultaneous movement of the right arm and of the torso. It should be noted here that, since ballistic movements are by definition not affected by external interferences, the training can be performed without considering the surrounding environment, as well as vision and auditory information. The task of the robot is to throw a small cube of side size 2 cm and weight 40 grams as far as possible according to an externally given velocity for the movement. The robotic model used for the experiments presented here is a simulation of the iCub humanoid robot, that was developed with the aim to accurately reproduce the physics and the dynamics of the physical iCub using a software library that provides an accurate simulation of rigid body dynamics and collisions.

Figure 2(a) presents the three action phases: (left) *Preparation* phase, the object is grabbed and shoulder and wrist joints are positioned at 90 degrees; (center) *Acceleration* phase, the shoulder joint accelerates until a

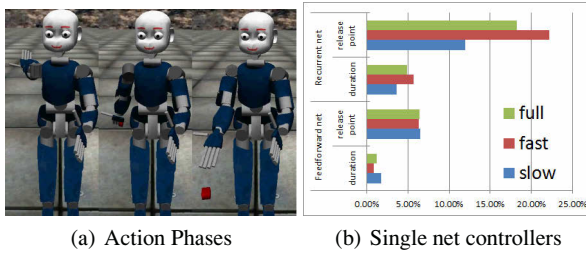


Figure 2: Action phases and performance comparison (by means of duration and release point errors) of FFNN and RNN as controllers with full range training

given angular velocity is reached, while the wrist rotates down; (right) *Release* phase: the object is released and thrown away. The experiments are divided into two phases: In the first phase the RNN was trained by a simple heuristic to predict its own subsequent sensorimotor state. To this end joint angle information over time was sampled in order to build 20 input-output sequences corresponding to different directions of the movement. In addition, in order to model the autonomous throw of an object, the primitive action to grab/release was also considered in the motor information fed to the network. In the second phase the RNN operates in *offline* mode and, thus, its prediction are made according only to the internal model built during the training phase.

### Experimental results and Discussion

In our experiments we tested the impact of mental training in action performance in a different speed range that was not experienced before. To this end we split both the learning and testing dataset into two subsets according to the duration of the movement: *fast* range subset comprises examples that last less than 0.3 seconds; *slow* range subset comprises all the others. Figure 2(b) shows the performance of the two nets as controller of the ballistic task. As expected the FFNN is the best controller for the task if the full range is given as training, thus, it is the ideal controller for the task.

To test the mental training, we compared results on three different case studies: (a) **full range**: For benchmarking purposes, it is the performance obtained by the FFNN when it is trained using the full range of examples (*slow* + *fast*); (b) **slow range only training**: The performance obtained by the FFNN only when it is trained using only the *slow* range subset. This case stressed the generalization capability of the controller when it is tested with the *fast* range subset; (c) **slow range plus mental training**: In this case the two architectures operate together as a single hierarchical architecture, in which first both nets are trained with the *slow* range subset, then the RNN runs in *mental imagery mode* to build a new dataset of *fast* examples for the FFNN, that is incrementally trained this way.

Results show that generalization capability of the RNN

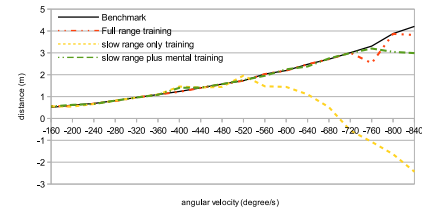


Figure 3: Results: Distance reached by the object after throwing movements of varying velocities. Negative values represents the objects falling backward;

helps to feed the FFNN with new data to cover the *fast* range, simulating *mental* training. In fact, the FFNN, trained only with the *slow* subset is not able to foresee the trend of duration in the *fast* range, this implies that *fast* movements last longer than needed and, because the inclination angle is over 90 degrees, the object falls backward (see Figure 3).

The FFNN failure in predicting temporal dynamics is explainable by the simplistic information used to train the FFNN, which seems to be not enough to reliably predict the duration time in a faster range, never experienced before. On the contrary, the greater amount of information that comes from the proprioception and the fact that the RNN has to integrate over time those information in order to perform the movement, makes the RNN able to create a sort of internal model of the robot's body behavior. This allows the RNN to better generalize and, therefore guide the FFNN in enhancing its performance.

### Conclusion

The results presented in this work, in conclusion, allow to imagine the creation of novel algorithms and cognitive systems that implement even better and with more efficacy the concept of artificial mental training. Such a concept appears very useful in robotics, for at least two reasons: it helps to speed-up the learning process in terms of time resources by reducing the number of real examples and real movements performed by the robot. An interesting direction for future work is the integration of the artificial imagery with reinforcement learning techniques, with the aim to improve the learning phase replacing real actions with *mental* simulations.

### Acknowledgment

The work was partially funded by the European Project ROBOT-ERA.

### References

- Di Nuovo, A. G., Marocco, D., Di Nuovo, S., and Cangelosi, A. (2013). Autonomous learning in humanoid robotics through mental imagery. *Neural Networks*, 41:147–155.



# Evaluation of an Experimental Framework for Exploiting Vision in Swarm Robotics

Sjriek Alers, Bijan Ranjbar-Sahraei, Stefan May, Karl Tuyls and Gerhard Weiss

Department of Knowledge Engineering  
Maastricht University

Email: {sjriek.alers,b.ranjbarsahraei}@maastrichtuniversity.nl,  
stefan.may@student.maastrichtuniversity.nl, {k.tuyls, gerhard.weiss}@maastrichtuniversity.nl

## Abstract

Visual feature detection with limited resources of simple robots is an essential requirement for swarm robotic systems. Robots need to localize their position, to determine their orientation, and need to be able to acquire extra information from their surrounding environment using their sensors, while their computational and storage capabilities might be very limited. This paper evaluates the performance of an experimental framework, in which environmental elements such as landmarks and QR-codes are considered as key visual features. The performance is evaluated for environmental light disturbances and distance variations and feature detection speed is thoroughly examined. The applicability of the approach is shown in a real robot scenario by using e-puck robots. Finally, the results of applying the approach to a completely different setting, i.e., simulation of pheromones using glowing trail detection, are presented. These results indicate the broad applicability range of the developed feature detection techniques.

## Introduction

In recent years there has been a rapidly growing interest in using teams of mobile robots for achieving complex tasks such as environmental coverage (Cortes et al., 2004) and exploration (Burgard et al., 2005). This interest is mainly motivated by the broad spectrum of potential civilian, industrial and military applications of multi-robot systems. Triggered by this interest, today, development of practical approaches for multi-robot problems is a well established topic in multi-robot research (e.g., (Hennes et al., 2012; Ranjbar-Sahraei et al., 2013)).

A natural phenomenon with high relevance to practically applicable multi-robot approaches is the foraging behavior of ants. In ant foraging, ants deposit pheromones on their path, while they are looking for either food or nest, which in long term establishes a path between these two locations (Dorigo et al., 2000). A slightly different foraging behavior can be seen among honeybees. Instead of using pheromones to navigate through an unknown environment, honeybees use a strategy called *Path Integration*, in combination with landmark navigation. These strategies turn out to be highly effective in solving distributed optimization problems (Lemmens, 2011). Although investigation of foraging behavior

of ants and bees is very interesting, the task of locating and acquiring resources in an unknown environment is quite a difficult task in practice (in particular with robots that have limited resources). Considering that the foraging task can be seen as an abstract representation for many other advanced tasks, such as patrolling and routing. A successful embodied implementation of distributed foraging can result in promising applications in, e.g., security patrolling, monitoring of environments, exploration of hazardous environments, search and rescue, and crisis management situations.

Getting motivation from the mentioned potential applications of distributed coordination and following the previous work (Alers et al., 2011; Lemmens et al., 2011), which mainly was relied on random exploration methods and infrared sensor data for obstacle detection, authors have recently introduced a framework for simple swarm robotic systems, which exploits vision in robots with very limited resources to extract information from landmarks and environmental patterns (Alers et al., 2013). These features are used as waypoints to navigate in an unknown environment, locate other entities, and detect modifications made in the environment. Although the previous paper describes the framework in detail, its performance in different environmental conditions, and various scenarios is not studied/compared yet. Therefore, this paper focuses on evaluation of this framework in different environmental settings (various light intensities, and detection distances). The detection speed is also deeply studied in various scenarios. In parallel, a new environmental feature, the glowing trails, is described; the developed approach is adapted to this feature, and the results are illustrated.

The remainder of the paper is structured as follows: First related work is briefly reviewed, then the vision-based framework for robots with limited resources is introduced. As the main contribution of this paper we address the influence of external variables, like environmental lighting conditions and viewing angles, on the detection performance of specific features. Also the reliability of detection of the various features are evaluated, and the overall performance regarding time and memory consumption in respect to us-

ability in real robot scenarios is investigated. Afterwards, the newly introduced feature, the glowing trails, and its application is described. Finally, two different types of swarm robotic implementations of this framework are illustrated, which can also be found online in (SwarmLab, Maastricht University, 2013a,b). The concluding remarks are included at the end of the paper.

### Related Work

Robotic systems use vision for accomplishing various tasks ranging from mobile robot navigation (DeSouza and Kak, 2002) to industrial applications (Gonzalez and Safabakhsh, 1982). However, most of the research in this field is focused on the image processing techniques which need a centralized unit to deal with the computations and memory storage tasks. For instance, (Winters et al., 2000) used a Pentium II 350MHz PC and (Chen and Birchfield, 2006) took advantage of a Dell Inspiron 700m laptop with 1.6 GHz CPU for vision-based mobile robot navigation. (Baeten and De Schutter, 2002) used a vision-based approach for accurate and fast task execution, while all of the computations were carried out by a digital signal processing module, serialized with a host computer.

When vision is required for decentralized units (e.g., robots in a swarm) either, each agent is equipped with relatively powerful resources (e.g., in (Quinlan et al., 2003) the Sony's AIBO robots are equipped with 576 MHz CPU and 64 MB RAM), or has a centralized unit with an overhead camera which processes the image, and sends the required data to the robots (e.g., in (Ranjbar-Sahraei et al., 2012a) a host computer denotes the exact position of robots). Alternatively, (Slusny et al., 2009) used a swarm of e-puck robots in which each robot takes an image individually, sends it to a centralized processing unit via Bluetooth, and receives the required data back from that server.

In contrast to the above mentioned works, being able to process images on a robot with very limited resources is a mandatory requirement for swarm robotic systems. In these systems using centralized units is impossible (due to the complexity and high amount of data). Equipping robots with high capabilities can be very expensive, although recent development of mobile phones can argue against this. Simple micro controllers will always be more cost effective. Therefore, in this paper we use the e-puck robot camera and its internal resources (i.e., a 60 MHz CPU and 8KB RAM) for detection of different features in the environment (e.g., barcodes, and QR-codes).

### A Vision-based Framework for Swarm Scenarios

In (Alers et al., 2013) we explored several visual features that can be used for acquiring information from the environment by a robot with limited computational abilities, equipped with a camera. For detecting key locations in the environment (e.g., corners in a maze), we investigated the

usage of specific landmarks for these locations. Each landmark consists of an upper ring with a solid color, so that it can be detected from a distance, and on the lower part a unique barcode for keeping track of the landmark numbers, as can be seen in Fig. 1a. Furthermore, we explored the possibility to detect markers with an even higher data density: QR-codes, as in Fig. 1b. The challenge in the detection of these two-dimensional codes, lies in analyzing and processing the camera data with the limited processing and memory resources that are available in our robotic platform. Finally, we explored the most common feature already available in every swarm robotic setting: the presence of other robots. It's always favorable to detect the relative distance and orientation to other robots in respect of one's position. Therefore, the available LEDs on the robot provide a very good feature for robot detection from a distance, see Fig. 1c. Moreover, we designed a specific gradient pattern for nearby robot detection, as shown in Fig. 1d, which can result in a very accurate orientation and distance detection.

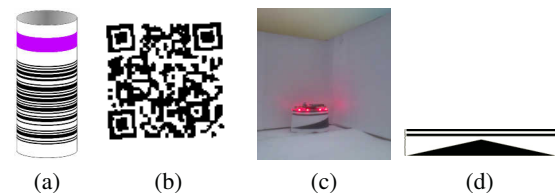


Figure 1: Detectable features presented in (Alers et al., 2013) (a) Landmarks with barcode. (b) QR-code level 3. (c) Robot LEDs. (d) Robot orientation pattern.

### Performance Evaluation

In this section the features, which were briefly addressed in the previous sections (described in more detail in (Alers et al., 2013)), are evaluated for their usability in real-world settings.

We start by evaluating several image filter techniques that are used to transform the captured image into a more suitable format. After this transition we run several utility functions, to cluster pixels or detect specific patterns. These filters and utility functions are the very basis of the feature detection and will run as a pre-processing step on every captured frame before the image is passed through to the actual feature detectors. Then we will describe the environmental influences and corrections needed for optimal performance under different circumstances. We also evaluate the detection performance based on distance variations of the detectable objects. Finally we give an overview of the time that is needed to detect each feature.

### Filter and Utilities

To see how applicable the filters are in a real robot setup it is important to know whether the filters can be used in

real-time for the specific setup, and whether for this particular setup all filters can be implemented due to the memory constraints of the platform. Therefore, in this subsection we provide time and memory allocation measurements of the provided filters, such as grayscale, Hue, several forms of Halftoning, and Gaussian blur filters. We also provide this information for the (group detection), (pattern finding) and (Hough transformation) utilities, which are described in more detail in (May, 2013). Furthermore, we provide overall detection-time measurements of the robot camera, that is needed to capture images.

**Filters** The various types of filters are the core elements of our feature detection. During the detection of a specific feature, often several filters are needed, and sometimes a single filter is used multiple times in one process. To measure the performance and increase the time measurement accuracies, we run the filters 1000 times repeatedly on one picture, and then divide the overall time by 1000 which gives us the running time for one specific filter. As the filters work independent of the data there should be no influence of the actual image to the measurement results. In Table 1 the average time that each filter needs to process a single image with a resolution of 40x40 pixels is listed, this table also lists the memory requirements for each filter. Looking at the memory requirements it should be noted, that only 8000 bytes are available, and a simple filter iteration can already consume a considerable amount of memory (e.g., a single run of the Halftoning histogram filter requires 3.4% of the total memory).

Filter	Time	Memory
grayscale	17.9ms	6 Byte
Hue filter	31.7ms	20 Byte
Halftoning threshold	3.3ms	2 Byte
Halftoning average	5.9ms	8 Byte
Halftoning midpoint	6.9ms	8 Byte
Halftoning histogram	18.8ms	272 Byte
Gaussian blur	24.4ms	7 Byte

Table 1: Resources used by filters

As can be seen in Table 1, the Hue filter is the most time consuming, this is due to the fact that it converts every pixel to the HSL colorspace. The Halftoning histogram filter is the most memory consuming. On the other hand the Halftoning filter with a specific threshold is the fastest and lowest memory consuming filter, as it needs no pre-calculation and can directly process the image.

**Utilities** The performance of the utilities described in (May, 2013), are more difficult to measure, as they depend on a various number of parameters. Therefore, each utility is evaluated in a separate part. Similar to filters, all utilities,

except for the Hough transformation, run 1000 times on a single image to make a more accurate time measurement.

## I Group Detection

The group detection algorithm locates all clusters of pixels that are non black, and determines their groups center points. The run-time of this algorithm depends on the amount of clusters that are detected, and on the amount of pixels that are included in a group. As can be seen in Table 2, the performance is highly dependent on the number of pixels per group. All processed images have a resolution of 40x40 pixels.

Groups	Pixel per group	Time	Memory
1	1	4.1ms	15 Byte
	9	6.1ms	41 Byte
	25	10.8ms	123 Byte
2	1	4.9ms	15 Byte
	9	8.8ms	41 Byte
	25	16.1ms	123 Byte
3	1	5.3ms	15 Byte
	9	12.5ms	41 Byte
	25	21.1ms	123 Byte
10	160	398.6 ms	15 Byte

Table 2: Resources used by group detection algorithm

Based, on the results in Table 2, this group detection algorithm should only be used in situations, where time is not a critical factor, as we usually deal with detection of about 10 groups.

## II Pattern Finder

There are two pattern characteristics which can influence the performance of the pattern finding algorithm; first the length of the pattern, and second the number of reoccurrences of the pattern in the image. Due to the fact that the pattern finder searches only a single column or row at each iteration, we have to make sure that this row or column is as long as possible. According to the specifications of the camera, the maximum length in vertical alignment is 480 pixels and in horizontal alignment 640 pixels. The latter direction is chosen for the performance measurements. The pattern itself has no influence on the performance, as the image is stored in run-length encoding, but the number of color changes in the pattern do influence the results.

As can be seen in Table 3, the time and the memory consumption mainly depend on the size of the pattern. The second parameter gives the reoccurrence of the pattern in the detection area. In practice this means that the algorithm needs up to 42.3ms, to scan every line of a halftone image with 96x96 pixel. From this we can

Pattern length	Matches	Time	Memory
3	1	2.8ms	33 Byte
	3	2.9ms	33 Byte
	5	2.9ms	33 Byte
9	1	4.0ms	45 Byte
	3	4.0ms	45 Byte
	5	4.1ms	45 Byte
15	1	4.8ms	57 Byte
	3	5.0ms	57 Byte
	5	5.1ms	57 Byte

Table 3: Resources used by pattern detection algorithm

conclude that it is possible to process even large images in an appropriate amount of time.

### III Hough Transformation

We implemented two versions of the Hough transformation, the standard one and the fast transformation. The standard Hough transformation is a time consuming algorithm, which can detect multiple lines. Due to the limitations of the memory of the e-puck, the classical Hough Transformation can not run at the same precision as the Fast Hough transformation, which can only detect one line. Unlike the other utilities presented, the Hough transformation is independent of the image data, it always needs the same amount of time and memory. All tests were done with a grayscale image of 40x40 pixel.

Algorithm	Time	Memory
Hough Transformation	2721.3ms	1800 Byte
Fast Hough Transformation	121.2ms	180 Byte

Table 4: Resources used by Hough Transformation

As can be clearly noted from Table 4, the Fast Hough transformation, reduces the run-time requirements by a factor of approximately 22, and the Memory requirements with a factor 10.

**Camera** In addition to the different filters and utilities, the performance and speed of the camera itself, on the overall performance of the system is measured. Pictures are captured in color, grayscale and halftone mode, and the capture time is computed by measuring the overall time required for capturing 1000 images and calculating the average as shown in Table 5. For color and grayscale images the same picture size is chosen. The advantage of halftone images is their higher data resolution, to accommodate this the resolution of the halftone pictures is higher than the one for color and grayscale.

Just capturing a single image takes much more time than running a single algorithm needed to process the image.

Mode	Size	Time	Memory
Colored	40x40 pixel	153.1 ms	3200 Byte
Grayscale	40x40 pixel	80.7 ms	1600 Byte
Halftone	96x96 pixel	1734.2 ms	2752 Byte

Table 5: Resources used by camera for different modes (i.e., color, gray-scale, halftone)

This can be a downside when a situation requires higher capture details, which can be done by taking multiple images and adding the processed data together. This will lead to higher accuracy, but at the cost of processing time.

### Environmental Light Condition

Light is a part of the environment which is uncontrollable in the real world. Depending on where, and how bright the light sources are, the images taken by the robot differ in their contrast and brightness. The robot's internal camera calibration tries to eliminate the influence of the light, and tries to return the same image for different brightnesses. With decreasing light intensity, it is more complicated for the auto correction to fulfill its task. To evaluate the light conditions we have built a box, which has white walls and is closed on all sides except for the top. This box, as can be seen in Fig. 1c, contains the robot, and the top is closed with a laptop monitor. The monitor acts as a controllable light source and has a constant light distribution of 300 cd/m<sup>2</sup>. In all our experiments we place a detectable feature inside this box and decrease the light intensity in 10 steps of 30 cd/m<sup>2</sup>, from full brightness till complete darkness.

**Landmark** In the first test landmarks are placed in the box. As shown in Fig. 2, the brightness of images is not that much influenced by the light condition. The first five images have almost the same brightness, but the noise density increases. Only in the last image, when the light intensity is completely reduced, the landmark is not recognizable any more.

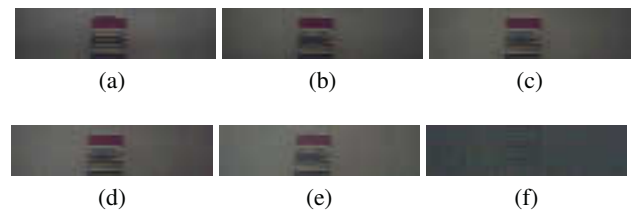


Figure 2: Detection of landmark for different brightnesses (a) 100% (b) 80% (c) 60% (d) 40% (e) 20% (f) 0%

From the histogram values in Fig. 3 it can be seen that the colors are all moving to the same area. This effect is due to the automatic light gain correction of the e-puck camera. It



influences the brightness in a way, that the average value of all pixels is always the same.

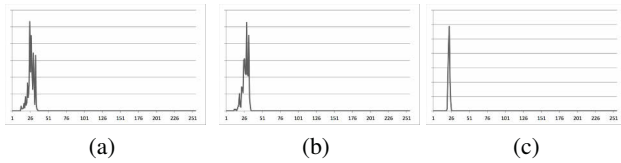


Figure 3: Grayscale histogram of landmark with brightness of the environmental light at (a) 100% (b) 60% (c) 0%

**Robot Detection** In order to examine how the e-puck camera handles the LEDs of another e-puck, we show in Fig. 4 the influence of environmental light on a captured image. Unlike in the images of the landmarks in previous experiment, there are significant differences in these images of this experiment. At 100% brightness, the LEDs cannot be differed from the surrounding environment (see Fig. 4a), but the body pattern is clearly detectable. With low light conditions, as can be seen in Fig. 4f, the LEDs are clearly distinguishable from the remaining image, but the body pattern cannot be detected anymore.

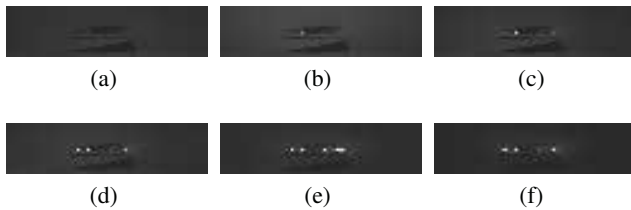


Figure 4: Images taken from an e-puck with brightness at (a) 100% (b) 80% (c) 60% (d) 40% (e) 20% (f) 0%

This effect can also be seen, when the camera zooms in into one of the single LEDs, that was detected in the overall scene. In Fig. 5a the white part of the LED fills only a small part of the image and the single LEDs can be clearly distinguished, while in Fig. 5f the two LEDs are forming one big part.

The influence of the environmental light on the detection of LEDs is very important. In bright environments the probability of detecting this feature is much smaller. In a dark environment, other features, like the body pattern, cannot be detected anymore. An optimal condition could be a light setting similar to the one in Fig. 4d, where the LEDs can be clearly differentiated from the environment and the body pattern still can be detected.

### Feature Distance

In this subsection, we address how reliable the e-puck can detect specific features, and how the detection rate is influenced by the distance to a specific feature. For this test,

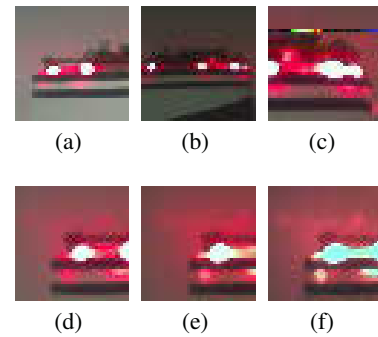


Figure 5: Zooming in to a detected LED of an e-puck with brightness at (a) 100% (b) 80% (c) 60% (d) 40% (e) 20% (f) 0%

we placed the robot in a white environment in front of only one detectable feature. The e-puck has to detect the feature and calculate its estimated position. The test is repeated 100 times with different angles and distances to the feature.

**Color Block** In Table 6 the results of running the color block detection algorithms are shown. During the test distance varies from 20cm to 80cm. Distances below 20cm are not taken into account, because the colored block is not visible in such distances.

Distance	Detection rate
20 cm	96 %
25 cm	97 %
30 cm	96 %
35 cm	96 %
40 cm	89 %
45 cm	78 %
50 cm	79 %
55 cm	68 %
60 cm	51 %
65 cm	38 %
70 cm	15 %
75 cm	7 %
80 cm	1 %

Table 6: Test results for colored block of landmarks

**EAN-8 codes** The EAN-8 code is a part of a landmark. As the colored block is also part of the landmark, the EAN-8 detection can benefit from this and it's relative good detection rate. Each time a purple block is detected the robot can be sure that the EAN-8 barcode is located below this block. However, the exact range of the EAN-8 code still has to be detected. The correctness of detecting and decoding the barcode depends on the distance from which the e-puck reads the EAN-8 code, as can be seen in Table 7. Values below

20 cm are not tested as this is the minimum distance to see the whole EAN-8 bar code. Table 7 also shows the correctness of the data after decoding, which is much lower than just detecting a valid pattern.

Distance	Detection rate	Correctness
20 cm	93 %	68 %
25 cm	89 %	58 %
30 cm	70 %	32 %
35 cm	58 %	5 %

Table 7: Test results for EAN-8 codes of landmarks

**QR-Codes** QR-codes provide high density information, but are quite complex to read. For example, one dimensional bar codes just have to be scanned in a line over the image, two-dimensional codes have to be transformed so that they fit in a fixed rectangle, only then it is possible to process them further. Hence, the detection of the outer shape is very important, but also often fails. The higher the QR-version, and thus the number of modules inside the code, the more exact the shape has to be determined. In Table 8 the detection rate of different versions at their optimal distances are listed. All measurements are done from a position right in front of the pattern. The error correction level is set to *H*, the highest possible level. The correctness is only determined in the cases where the code was correctly detected.

**Robot Detection** In practice, the detection of other robots is done with two different algorithms. First the program tries to detect the body pattern of the robot. When this algorithm does not detect any robot, robot localization bases on the LEDs is performed. In Table 9 the detection rate at different distances is listed, as well as the distance estimation. As the LED based detection does not return any distance estimation the value in the third column is only calculated if the e-puck is detected by the body pattern recognition algorithm.

## Overall Performance

Considering all of the in-detail examination, the performance of a feature detection algorithm is a combination of the processing time of the involving filters, utilities, and also image capturing time. The exact time depends on how often each step has to be executed, and if the objects are recognized by the detection algorithms. In Table 10 the overall processing times for the different features are presented.

Version	QR code size	Detection rate	Correctness
1	21x21	80 %	53 %
2	25x25	48 %	5 %
3	29x29	23 %	0 %

Table 8: Light tests of QR-code detection

Distance	Detection rate	Distance estimation
10 cm	100 %	95 %
15 cm	100 %	79 %
20 cm	75 %	60 %
25 cm	84 %	—
30 cm	58 %	—
35 cm	61 %	—
40 cm	52 %	—
45 cm	39 %	—
50 cm	20 %	—
55 cm	9 %	—

Table 9: Distance tests of Robot detection

Algorithm	Time
Colored block	204.3 ms
EAN-8 code	324.1 ms
Single LED	900.0 ms
Three LEDs	2334.8 ms
Body pattern	198 ms
QR-code	3253.0 ms

Table 10: Required time for detection of the features

From these overall feature detection times we can conclude that detecting static objects such as colored blocks and EAN-8 codes in a real swarm robot scenarios is doable. Even detecting a QR-code is possible, as long as the location of this code in the environment is known, as searching for a QR-code, using vision, requires high processing time, and memory.

## Glowing Trails

As an extension to our current framework we introduce a new feature, the glowing trail. Inspired by nature, in which insects use chemicals for indirect communication (known as Stigmergy), researchers are interested in applying stigmergy in multi-robot systems, as well. However, in-field deployment of an indirect communication requires manipulating environment which is not a trivial task. Researchers have recently used a few techniques for accomplishing this task. For instance, chemical materials has been proposed by (Fujisawa et al., 2008). Due to difficulties in implementation and limited extendibility, this approach didn't provide sufficient applicability in swarm scenarios.

As an alternative, glowing trails exploited by (Kronemann and Hafner, 2010), inspired on (Alers and Hu, 2009), and further extended for swarm robotic scenarios by authors (Ranjbar-Sahraei et al., 2013) are easy to set up in laboratory environment and still very efficient. These glowing trails can help robots to communicate indirectly to achieve their goals (e.g., environmental coverage, intruder tracking, etc.). A simple indirect communication is shown in Fig. 6 in which robots announce their territory border by putting

pheromones on the borders (Ranjbar-Sahraei et al., 2012b).

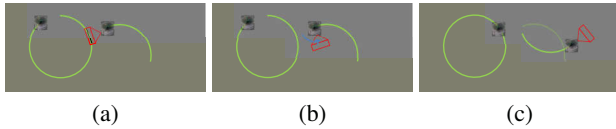


Figure 6: Using stigmergic communication for efficient area coverage proposed by (Ranjbar-Sahraei et al., 2012b) (a) robot on right hand side detected the glowing trail. (b) robot changes its circling direction. (c) robots establish separate territories

For detection of the glowing trails, in contrast to the simple method used in (Kronemann and Hafner, 2010), in which photo-sensors were used to detect glowing trails, the e-puck vision approach as described above can be used. Therefore, we take advantage from the developed techniques for color filtering and pattern recognition which are designed based on the limited resources of an e-puck robot, and still powerful enough to extract information from the trails.

The new glowing trail feature can be seen in Fig. 7. The detected grayscale image is converted to a black-white image with a fixed threshold. The amount of white pixels is determined to see if there is any trail in the image. When the image has more than 1% of white pixels, a Fast Hough Transformation (Gonzalez and Woods, 2002) is performed to determine the direction of the trail, see Fig. 7(b).

For using glowing trails, the floor should be covered by phosphorescent material which absorbs UV light and re-emits the absorbed light at a lower intensity for up to several minutes after the original excitation. Robots should also be equipped with UV-LEDs to emit light to the glowing materials.



Figure 7: New introduced feature: (a) Initial image from a glowing trail received by e-puck camera. (b) Filtered image with a red directional line determined by Hough-Transformation (May, 2013)

### Real World Evaluation

To test the proposed features and algorithms under real conditions we used this framework to implement two different swarm approaches. One approach focuses on using these features and algorithms in path optimization problems as in (Alers et al., 2011), the other swarm approach focuses on area coverage with glowing trails as in (Ranjbar-Sahraei et al., 2012b).

For the path optimization approach we described and implemented our framework in (Alers et al., 2013). We ex-

amined the proposed approach in a real scenario, in an environment as shown in Fig. 8. A video of this performed experiment can be found online in (SwarmLab, Maastricht University, 2013b), including the intermediate image data from the robot.

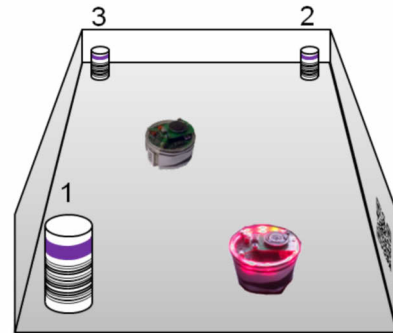


Figure 8: Scenario for validation of proposed approach

For the coverage approach, we used the new glowing trails framework extension, introduced in previous section. In (Ranjbar-Sahraei et al., 2013), we demonstrated an implementation of this approach, the results of applying this vision-based trail detection on a real swarm of e-puck robots is shown in Figs 9a- 9c. A video of the performed experiments can be found in (SwarmLab, Maastricht University, 2013a).

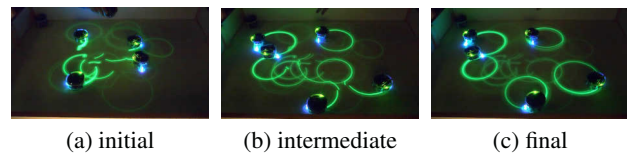


Figure 9: Vision-based detection of glowing trails approach

### Discussion, Conclusions & Future Research

In this paper we presented an in-depth study of a vision-based feature detection framework for multi-robot scenarios. This study covered a complete range of performance evaluations, ranging from measuring detection rate for different environmental brightnesses and detection distances to detection accuracy for different features. Furthermore, various experiments on a real e-puck robot were performed to measure the required time for different tasks such as applying gray-scale filter, halftone filter, group detection, and pattern finding algorithms.

From the overall performance measurements we can conclude that detection of objects that are not too complex is easily doable. However, the detection time, and the required memory increases drastically when more complex objects are chosen as features. Moreover, we showed that the environmental light variation doesn't affect the detection of the

features, except in the extremes when it is either too bright or too dark. The detection of other robots via their LEDs is doable in a static situation, but in most swarm scenarios the robots move continuously in the environment, which makes detection of moving robots an open research question in our research. Moreover, for the newly introduced glowing trail feature we demonstrated a working dynamic multi-robot scenario, which encourages us to investigate applications in dynamical swarm-optimization settings. Finally, we as the main future work, we are working on integration of the proposed techniques with the bee-inspired foraging algorithms, where the big challenge in this is to fit all foraging and vision algorithms within the limited memory of the robot.

## References

- Alers, S., Bloembergen, D., Hennes, D., de Jong, S., Kaisers, M., Lemmens, N., Tuyls, K., and Weiss, G. (2011). Bee-inspired foraging in an embodied swarm (demonstration). In *Proceedings of the Tenth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2011)*, pages 1311–1312.
- Alers, S. and Hu, J. (2009). Admoveo: A robotic platform for teaching creative programming to designers. In *Proceedings of the 4th International Conference on E-Learning and Games: Learning by Playing. Game-based Education System Design and Development (Edutainment 2009)*, Edutainment '09, pages 410–421, Berlin, Heidelberg. Springer-Verlag.
- Alers, S., Ranjbar-Sahraei, B., May, S., Tuyls, K., and Weiss, G. (2013). An experimental framework for exploiting vision in swarm robotics. In *ADAPTIVE 2013, The Fifth International Conference on Adaptive and Self-Adaptive Systems and Applications*.
- Baeten, J. and De Schutter, J. (2002). Hybrid vision/force control at corners in planar robotic-contour following. *Mechatronics, IEEE/ASME Transactions on*, 7(2):143–151.
- Burgard, W., Moors, M., Stachniss, C., and Schneider, F. E. (2005). Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, 21(3):376–386.
- Chen, Z. and Birchfield, S. T. (2006). Qualitative vision-based mobile robot navigation. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2686–2692. IEEE.
- Cortes, J., Martinez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255.
- DeSouza, G. N. and Kak, A. C. (2002). Vision for mobile robot navigation: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(2):237–267.
- Dorigo, M., Bonabeau, E., and Theraulaz, G. (2000). Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16(8):851–871.
- Fujisawa, R., Imamura, H., Hashimoto, T., and Matsuno, F. (2008). Communication using pheromone field for multiple robots. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1391–1396.
- Gonzalez, R. and Woods, R. (2002). *Digital image processing*. Prentice Hall Upper Saddle River, NJ.
- Gonzalez, R. C. and Safabakhsh, R. (1982). Computer vision techniques for industrial applications and robot control. *Computer*, 15(12):17–32.
- Hennes, D., Claes, D., Meeussen, W., and Tuyls, K. (2012). Multi-robot collision avoidance with localization uncertainty. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 147–154. International Foundation for Autonomous Agents and Multiagent Systems.
- Kronemann, M. L. and Hafner, V. V. (2010). Lumibots making emergence graspable in a swarm of robots. In *DIS 2010, The ACM Designing Interactive Systems Conference*, pages 408–411, Aarhus. ISBN 978-1-4503-0103-9.
- Lemmens, N., Alers, S., and Tuyls, K. (2011). Bee-inspired foraging in a real-life autonomous robot collective. In *Proceedings of the 23rd Benelux Conference on Artificial Intelligence (BNAIC 2011)*, pages 459–460.
- Lemmens, N. P.-P. M. (2011). *Bee-inspired Distributed Optimization*. Maastricht University.
- May, S. (2013). E-puck vision: Detecting key features with limited resources. Master's thesis, Maastricht University.
- Quinlan, M. J., Chalup, S. K., and Middleton, R. H. (2003). Techniques for improving vision and locomotion on the sony aibo robot. In *Proceedings of the 2003 Australasian Conference on Robotics and Automation*.
- Ranjbar-Sahraei, B., Alers, S., Tuyls, K., and Weiss, G. (2013). Stico in action (demonstration). In *Proceedings of the twelfth International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*.
- Ranjbar-Sahraei, B., Shabaninia, F., Nemati, A., and Stan, S. (2012a). A novel robust decentralized adaptive fuzzy control for swarm formation of multiagent systems. *Industrial Electronics, IEEE Transactions on*, 59(8):3124–3134.
- Ranjbar-Sahraei, B., Weiss, G., and Nakisae, A. (2012b). A multi-robot coverage approach based on stigmergic communication. In *Multiagent System Technologies*, volume 7598 of *Lecture Notes in Computer Science*, pages 126–138. Springer.
- Slusny, S., Neruda, R., and Vidnerová, P. (2009). Localization with a low-cost robot. In *ITAT*, volume 584 of *CEUR Workshop Proceedings*, pages 77–80.
- SwarmLab, Maastricht University (2013a). Demonstration of an experimental framework for exploiting vision in stigmergic communication. <http://swarmlab.unimaas.nl/stico/indoor-experiments/>.
- SwarmLab, Maastricht University (2013b). Demonstration of an experimental framework for exploiting vision in swarm robotics. <http://swarmlab.unimaas.nl/papers/adaptive-2013-demo/>.
- Winters, N., Gaspar, J., Lacey, G., and Santos-Victor, J. (2000). Omni-directional vision for robot navigation. In *Omni-directional Vision, 2000. Proceedings. IEEE Workshop on*, pages 21–28. IEEE.



# Boolean Network Robotics as an Intermediate Step in the Synthesis of Finite State Machines for Robot Control

Lorenzo Garattoni<sup>1</sup>, Andrea Roli<sup>2</sup>, Matteo Amaducci<sup>2</sup>, Carlo Pinciroli<sup>1</sup> and Mauro Birattari<sup>1</sup>

<sup>1</sup>IRIDIA, Université Libre de Bruxelles, Belgium

<sup>2</sup>DISI-Cesena, *Alma Mater Studiorum* Università di Bologna, Italy  
lorenzo.garattoni@ulb.ac.be

## Abstract

We propose an approach to the automatic synthesis of robot control software based on the finite state machine (FSM) formalism. In our previous research, we have introduced Boolean network robotics as a novel approach to the automatic design of robot control software. In this paper, we show that it is possible to leverage automatically designed Boolean networks to synthesize FSMs for robot control. Boolean network robotics exhibits a number of interesting properties. Firstly, notwithstanding the large size of the state space of a Boolean network and its ability to display complex and rich dynamics, the automatic design is able to produce networks whose trajectories are confined in small volumes of the state space. Secondly, the automatic design produces networks in which one can identify clusters of states associated with functional behavioral units of the robots. It is our contention that the automatic design of a Boolean network controller can be a convenient intermediate step in the synthesis of a FSM, which offers the advantage of being a compact, readable, and modifiable representation. In this paper, we show that clusters of states traversed by network trajectories can be mapped to states of a FSM. We illustrate the viability of our proposal in two notable robotic tasks, namely collision avoidance and sequence recognition. The first task can be achieved by a memoryless control program, whilst in the second the robots need memory.

## Introduction

The automated design of compact high-level representations of control software for robots is a challenge in artificial intelligence. Through methods of automatic design, a robot learns a behavior without the explicit intervention of the developer. Automatic design methods offer advantages with respect to manual methods in terms of robustness and generality of the design process. Moreover, the space of solutions explored by automatic techniques is larger and less constrained than that explored by methods of manual design (Koza et al., 2003; Lipson, 2005). However, the effectiveness of automatic methods depends on a number of aspects such as the definition of the search space and the existence of a predictive simulation of the system. Yet, such methods do not provide guarantees on the solution optimality. Automatic design techniques act iteratively on the robot

control software in order to reach a configuration that fulfills the requirements. There exist several ways of representing control software of robots, but the finite state machine (FSM) formalism is the oldest and is broadly used.

Our intention is to propose a new way to automatically design FSMs representing control software for robots. The new approach exploits the work carried out in our previous research in the field of Boolean network robotics (Roli et al., 2011) as an intermediate step in the automatic design of FSMs. In the following, we first give an overview of the existing methods for the automatic design of robot control software and then we introduce the original contribution of our work.

Among the existing approaches to automatically obtain a FSM of a control software, evolutionary programming is one of the most notable (Fogel, 1962, 1993). EP is a paradigm used for the generation of programs, code, algorithms and structures in general, by means of variation and selection mechanisms inspired by natural evolution. Although EP was shown to produce interesting results in many important applications, several issues are still open about its employment (O'Neill et al., 2010). One of the main issues is the choice of the most appropriate representation for the programs to be evolved. In fact, the most suitable representation and the appropriate encoding of the programs into individuals in the evolution process are critical aspects for the performance of EP (Petrovic, 2007). Moreover, EP normally requires the definition of constraints to contain the size of the FSM.

Besides the evolution of FSMs, most of the effort in the field of automatic design of robot control systems has been concentrated around artificial neural networks (NNs) (Nolfi and Floreano, 2000). While NNs offer advantages such as high plasticity and adaptability, they are black-boxes and it is often very difficult to analyze their dynamics. The dynamical behavior of a NN can be modeled by a system of differential equations. There exists a number of works that show how the mathematical tools of dynamical systems theory can be used to gain significant insight into the dynamics of small continuous-time recurrent neural networks (Beer, 1995; Ya-

mauchi, 1993; Yamauchi and Beer, 1994). However, when the number of neurons is greater than a few units, the analysis becomes too complex to handle and only qualitative or approximate studies are possible.

In our previous work (Roli et al., 2011), we have introduced Boolean network robotics as a novel approach to the automatic design of robot control software. Boolean networks (BNs) are a model of genetic regulatory networks (Kauffman, 1969). BNs are extremely interesting from an engineering perspective because of their ability to display complex and rich dynamics, despite the compactness of their description and the simplicity of their implementation. BN dynamics can be studied through traditional dynamical system methods (Bar-Yam, 1997; Serra and Zanarini, 1990). The use of concepts such as state space, trajectories and attractors, combined with the discrete nature of BNs, enables non-trivial analysis of the dynamical behavior. Such ease of analysis is one of the strengths of BN systems.

In this paper, we continue the line of research in Boolean network robotics showing that the automatic design of a Boolean network controller can represent an intermediate step in the synthesis of a FSM. The intuition stems from the analysis we carried out on the trained Boolean networks and two interesting properties it revealed. First, the automatic design shapes the network dynamics in very limited volumes of the state space. Second, such dynamics are structured in sets of clusters of states associated with functional behavioral units of the robots. On the basis of such properties, we propose a heuristic to map those clusters into states of a FSM, which offers a compact, readable, modifiable, and formally verifiable representation.

In this work, we propose a proof of concept of our proposal applying Boolean network robotics to two robotic tasks, i.e., corridor navigation and sequence recognition. The first task is a typical collision avoidance behavior and consists in moving along a corridor avoiding walls and objects; this task can be attained by a robot equipped by a memoryless control software. Conversely, the second task presents a sequence-recognition scenario (Sun and Giles, 2001). The complexity of the target task lies in the fact that it requires the robots to have memory of the past in order to choose the next actions to perform.

The analysis of the trained networks confirms the two properties mentioned and allows for simple mapping between clusters of states in the state space of a BN and states of a FSM.

Despite the simplicity of the two tasks, this work represents a first crucial step towards the definition of an automatic design method of FSMs that exploits Boolean network robotics as a convenient intermediate step.

### Boolean network robotics

In this section we first introduce BNs and then we describe how they are employed and configured to let robots perform

the desired tasks.

### Boolean networks

A Boolean network is a discrete-state and discrete-time dynamical system. Its structure is defined by an oriented graph with  $N$  nodes each associated to a Boolean value  $x_i$ ,  $i = 1, \dots, N$ , and a Boolean function  $f_i(x_{i_1}, \dots, x_{i_{K_i}})$ , where  $K_i$  is the number of inputs of node  $i$ . The arguments of function  $f_i$  are the Boolean values of the nodes whose outgoing arcs are connected to  $i$ . The state of the system at time  $t$ , with  $t \in \mathbb{N}$ , is defined as the vector of the  $N$  Boolean values at  $t$ . The state space size is  $2^N$ . Several update schemes can be defined (Gershenson, 2004), but the most studied is characterized by synchronous and deterministic operations.

BN dynamics can be studied by means of the usual dynamical system methods (Bar-Yam, 1997; Serra and Zanarini, 1990), hence the usage of concepts such as state space, trajectories, attractors and basins of attraction. Recently, the attention of the scientific community has focused on the employment of efficient mathematical and experimental methods for analyzing network dynamics and thus have insight into the behavior of a BN system (Fretter and Drossel, 2008; Ribeiro et al., 2008; Serra et al., 2007).

### BN-Robot coupling

To design a BN-based robot control system, we first need to couple the BN to the robot so as to let the BN dynamics guide the robot behavior. For this purpose, some nodes of the network are given special roles. More precisely, we define a set of input nodes and a set of output nodes. This choice characterizes our approach with respect to most of the work performed about BNs, in which they are considered as isolated systems, even though some notable exceptions exist (Ansaloni et al., 2009; Dorigo, 1994; Kauffman, 1991; Patarnello and Carnevali, 1986). The Boolean values of the input nodes are not determined by the network dynamics, but they are imposed according to the robot sensor readings. Similarly, the values of the network's output are used to encode the signals for maneuvering the robot's actuators. Several ways to define the mapping between sensor readings and network's input, and between network's output and actuators are possible. However, the most natural way is to define the mapping via a direct encoding. Figure 1 shows the coupling between BN and robot.

### Automatic design methodology

Once a mapping between the BN and the robot is defined, the BN must be designed in order to control the robot's behavior. Our approach consists in treating BN design as a search problem. In fact, the design of a BN that satisfies given criteria can be modeled as a constrained combinatorial optimization problem by properly defining the set of decision variables, constraints and the objective function.

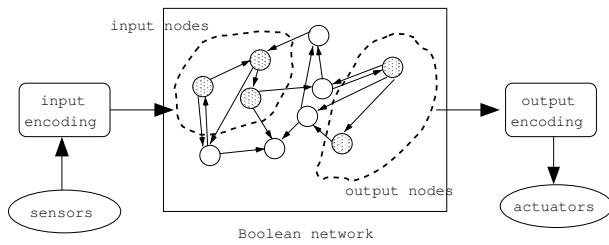


Figure 1: The coupling between BN and robot.

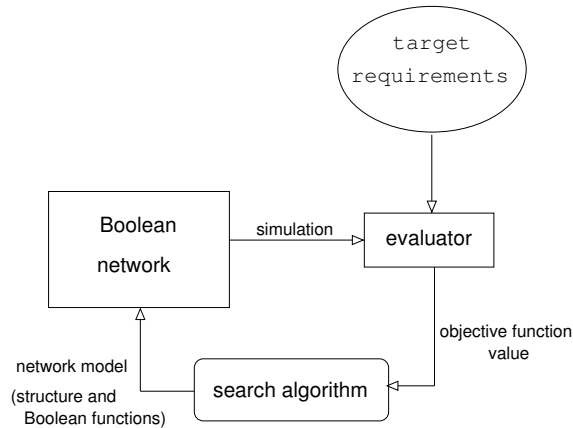


Figure 2: BN design process.

The search algorithm manipulates the decision variables which encode structure and Boolean functions of a BN. A complete assignment of these variables defines an instance of a BN. Then, we couple this network to the robot through the input-output mapping, and subsequently we execute the network. The evaluation of the network at each iteration of the search process is performed in a batch of simulated experiments. The performance of the robot in each experiment is assessed according to a user-defined objective function, which associates the robot behavior to a numeric evaluation. Finally, the search algorithm exploits this value of performance to proceed with the design process. In particular, the algorithm changes the configuration of the decision variables so as to find networks with better performances. This process is depicted in Figure 2.

### Robot tasks

We addressed two test cases with different characteristics: the first, corridor navigation, is a memoryless task while the second, sequence recognition, requires memory. It is interesting to analyze how the nature of the task influences the organization of the state space in the trained networks. In particular, our analysis aims to determine whether the state space of the trained networks exhibit the same properties independently of the nature of the task. These properties, which are (i) the compression of the dynamics in limited regions of state space, and (ii) the organization in clusters of

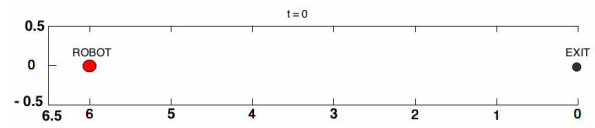


Figure 3: Corridor navigation environment.

states associated with behavioral units of the robots, enable the exploitation of the trained networks as intermediate step in the synthesis of FSMs.

In the remainder of this section we present the working environment and describe the two test cases.

### Robot and Simulator

For both test cases, the robots are trained in simulation. The simulation framework we employed is the open source simulator ARGoS (Pinciroli et al., 2012). ARGoS is a discrete-time, physics-based simulation environment that provides a faithful simulation of the behavior of different robotics platforms.

The robot simulated in our test cases is the *e-puck* (Mondada et al., 2009). The *e-puck* is a small wheeled robot, designed for research and educational purposes. It has a cylindrical body of 7 cm of diameter, equipped with a variety of sensors. For our test cases, we use the 8 infra-red proximity sensors placed along the circular perimeter of the robot and the 3 infra-red sensors pointed directly at the ground in front of the robot. The 3 latter sensors can be used to detect the color of the ground, in greyscale. The actuators utilized, besides the motors of the two wheels, are the 8 red LEDs.

### Corridor navigation

The first test case is designed to explore the features of networks able to perform a memoryless task. It consists of a robot that must navigate along a corridor avoiding any collision with the walls and finally reach the exit.

**Environment:** it consists of a straight corridor of 6.5 m in length and 1 m in width.

**Task:** at the beginning, the robot is placed within the corridor 6 m far from the exit. During the experiment, the robot must advance along the corridor, avoiding collisions and finally, within the given total execution time  $T = 120$  s, reach the exit. See Figure 3 for a representation of the environment at the beginning of the experiment.

During the execution, if a collision between the robot and the walls of the corridor occurs, the experiment is immediately stopped.

**Performance measure:** the performance assigned to the robot is simply its final distance from the exit (normalized). The smaller is this distance, the better is the performance of the robot.

**BN-robot setup:** for successful navigation, the robot needs the 8 proximity sensors to detect the walls and avoid them.

At each time step, the readings of the 8 sensors are encoded into the values of the BN input nodes. We use 4 input nodes to encode the readings of the proximity sensors. Thus, the 8 proximity readings are gathered in pairs. If at least one of the two sensors of the pair exceeds a chosen threshold, the corresponding input node value is set to 1. The pairs are formed to allow the robots to detect walls in the four directions north-east, south-east, south-west and north-west.

Once the readings of the sensors are encoded in the input nodes, the network's state is updated and finally the values of the output nodes are read, decoded and utilized to set the actuators. We use two output nodes to set the wheel speeds either to zero or to a predefined, constant value.

For this test case, we set the network size to 20 nodes. We leave the analysis on how this value affects performance for future investigation.

**BN design:** the initial topology of the networks, i.e., the connections among the nodes, is randomly generated with  $K = 3$  (i.e., each node has 3 incoming arcs) and no self-connections, and it is kept constant during the training. The initial Boolean functions are generated by setting the 0/1 values in the  $f_i$  uniformly at random. Our search process, which is a stochastic descent, works only on the Boolean functions. In particular, at each iteration, the search algorithm changes the configuration of the network by flipping one bit of the Boolean functions. The flip is performed by changing a random entry in the  $f_i$ , where  $i$  is a randomly chosen node. The new configuration is accepted if the corresponding BN-robot system has a performance at least equal to the current one. The evaluation of each network is performed on a set of initial conditions, that form the training set. For this test case, the training set is composed of six different initial orientations of the robot. The six angles are chosen so as to have six equally spaced orientations in the range between  $\frac{\pi}{3}$  and  $-\frac{\pi}{3}$  (with 0 that is the straight direction of the robot towards the exit). In this manner, the robot must be able to cope with a wide range of different situations and avoid the walls it detects in any direction. The final evaluation assigned to the robot is computed as the average of the performance across the 6 trials. We executed 100 independent experiments, each corresponding to a different initial network. In each experiment, we run the local search for 1000 iterations.

### Sequence recognition

The second test case aims to explore the properties of networks able to perform a task that requires memory. The task is sequence recognition (Sun and Giles, 2001). In particular, the robot must learn to recognize a sequence of colors by performing certain actions. This kind of task is more complex than the previous one, because the robot needs a form of memory to be able to choose the next action depending on the past.

**Environment:** it consists of a straight corridor of 7 m in

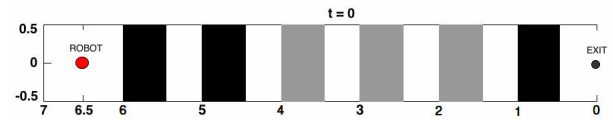


Figure 4: Sequence recognition corridor environment.

length and 1 m in width. Along the corridor, the ground is painted to form a striped pattern with three different colors: white (W) represents the background, while black (B) and gray (G) are the symbols of a sequence to be recognized.

**Task:** at the beginning of the experiment, the robot is placed within the corridor 6.5 m far from the exit. During the experiment, the robot must move along the corridor and reach the exit. Every time the robot encounters a black or gray area in the right sequence, it must turn its LEDs on. Conversely, when the robot encounters the background color or other colors in the wrong order, it must keep its LEDs off. The sequence to be recognized is a cyclic repetition of black followed by gray. By performing the right sequence of actions while moving along the corridor, the robot must be able to reach the exit within the given total execution time, fixed in  $T=130$  s. Figure 4 represents an example of the environment at the beginning of the experiment.

In the environment depicted in Figure 4, the robot must perform the following sequence of actions to achieve the goal (omitting the background color (W) whose corresponding LED correct status is always OFF):

Colors along the corridor					
B	B	G	G	G	B
ON	OFF	ON	OFF	OFF	ON
Robot's LEDs correct status					

If the robot, at any instant in time during the execution, performs the wrong action, the experiment is immediately stopped.

**Performance measure:** The performance assigned to the robot is the final distance from the exit of the corridor (normalized between 0 and 6.5). The value must be minimized.

**BN-robot setup:** for this task, the robot needs the ground sensor to detect the color of the ground. For our simple application we use only the central sensor. Since we encode three values (W, B, G), at each time step, the reading of the sensor is encoded into the values of two BN input nodes. We use four nodes to encode the proximity sensors that, even though not strictly needed for the task, can be still useful for the navigation along the corridor.

After the network's state update, we decode and use the values of the output nodes to set the actuators. Besides the two nodes used to control the wheel speeds, an additional output node is utilized to set the state of the LEDs either to ON or OFF.

For this test case we increased the network size to 30 nodes.



**BN design:** initial topology and Boolean functions are randomly generated with  $K = 3$ . In our experiments, the search strategy is a stochastic descent and works only on the Boolean functions, leaving the topology unchanged. The evaluation of each network is performed on a set of initial conditions. More precisely, the training set is composed of 10 different randomly generated sequences of colors on the ground. Differently from the corridor-navigation case, the robot starts always pointing towards the exit. In this way, the navigation task is simplified so as to focus the complexity on the sequence recognition. The final evaluation of a robot is the average value of the performance across the 10 trials. Due to the high computational cost required by each experiment, we executed only 30 independent experiments with 30 different initial networks for 100000 iterations of the search algorithm.

### Analysis of the results

The analysis of the results obtained in both test cases reveals two properties. First, the dynamics of the automatically designed networks spans across a very limited region of the whole potential state space. This means that the search algorithm moves towards networks whose dynamics are compact. This relationship between the design process and the dynamical features of the networks is notable: the search algorithm acts directly only on the network structures, searching for a good behavior of the BN-robot systems while ignoring the dynamics property of the networks. Nevertheless, the analysis shows that the algorithm shapes and compresses indirectly the dynamics of the networks.

The second property observed is the organization of the state space traversed by the final networks in a set of clusters of states, each devoted to perform a specific series of actions.

In the remainder of this section we present the analysis and the results obtained for both test cases.

#### Corridor navigation analysis

Once the design process is completed, the focus of the study is on the dynamical features of the resulting networks. The first aspect we analyzed is the measure of the fraction of state space utilized by the trained networks. In order to carry out this analysis, we collected a large number of trajectories, corresponding to different initial conditions, for each BN obtained. Then, we counted the number of different states that each network traversed across all the trajectories and we reported the empirical cumulative distribution of the resulting values. Figure 5 shows the distribution for the corridor-navigation test case.

The plot shows that the final network dynamics traverse limited volumes of the state space. In fact, the median usage of state space in the 100 trained networks is located around 150 states. This is a very tiny fraction of the whole potential space, whose dimension is  $2^N$  ( $2^{20}$  in this case). This first property enables an analysis of the network dynamics that

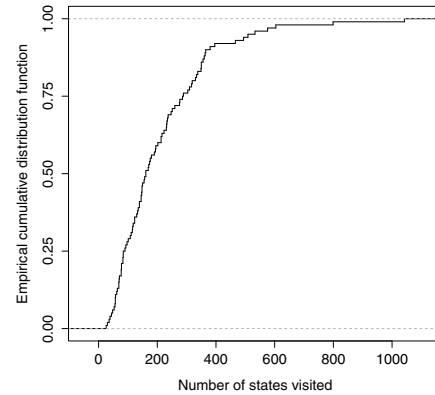


Figure 5: Empirical cumulative distribution function of the number of visited states in final networks. Corridor-navigation test case.

allows to gain significant insight into the behavior of the BN-robot systems.

To analyze the organization of the dynamics of a BN controlling a robot, we collected its trajectories by simulating the experiment. Then, we gathered the trajectories and we generated the graph of the observed state transitions. For lack of space, the graphs can be found as on-line supplementary material (Garattoni et al., 2013).

The state space of the robot performing corridor navigation can be decomposed in three macro areas. One is responsible of the behavioral units that react to walls detected on the east side of the robot. Likewise, another cluster of states is devoted to avoid the obstacles on the west side of the robot. Besides, the two areas are both connected to a third cluster, responsible of moving the robot straight ahead as long as no obstacle is detected. Furthermore, it is possible to observe that each cluster of states contains few topical states, visited many times, and a series of other nodes gradually increasing in number and decreasing in visits. To verify this property, we performed the analysis of the graph for all the final networks of the corridor-navigation test case. We report in a plot the cumulative distribution of the fraction of states visited at least  $v$  times, where  $v$  is the number of visits on the x-axis. The results, showed in Figure 6 for a typical case, suggest that the dynamical behavior of a BN is built around few, prominent states that correspond to the main traits of the robot behavior.

The observations and the analysis presented so far suggest a procedure for deriving a representation of the robot's dynamics in the form of a FSM. We determine the states of the FSM by starting the observation from the topical states and gradually moving to the less important ones. The result is that a state in the FSM takes the place of a clusters of connected states in the state space in which the BN remains until

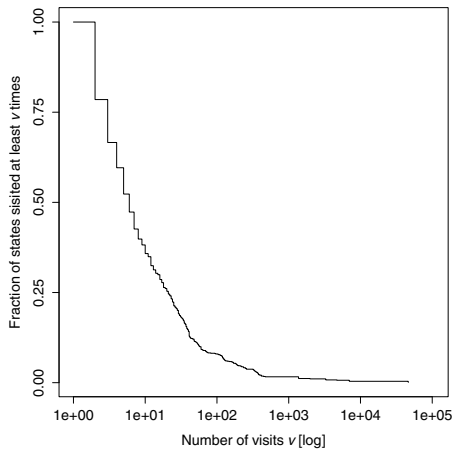


Figure 6: Distribution of the fraction of states visited at least  $v$  times. The fraction is computed with respect to the total number of states visited in 200 runs of 120000 time steps each. Corridor-navigation test case.

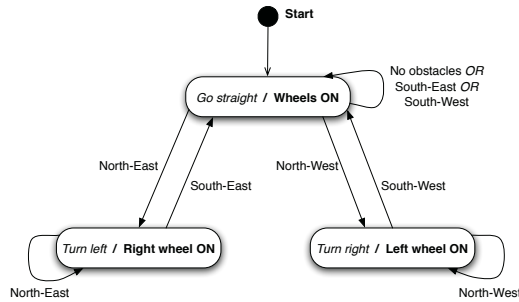


Figure 7: Finite state machine of the state space graph. Corridor navigation.

a specific input is received. By following this simple heuristic for a typical case of corridor navigation, we derived the FSM in Figure 7. We can observe that the automaton corresponds to a very simple yet effective behavior: the robot goes straight until an obstacle is detected on one side; in that case, the robot turns to the other side.

### Sequence recognition analysis

The analysis carried out for the second test case is similar to that presented for the corridor navigation test case. Due to the higher complexity of the sequence-recognition task with respect to the corridor navigation and the important computational cost required by each run, the number of successful networks to analyze is much lower than in the first test case. However, the analysis confirms the same properties.

The number of states visited by the trained networks is very low: the state space usage is on average 200 states out of  $2^{30}$  potential states. By collecting the trajectories of the successful networks and generating the correspond-

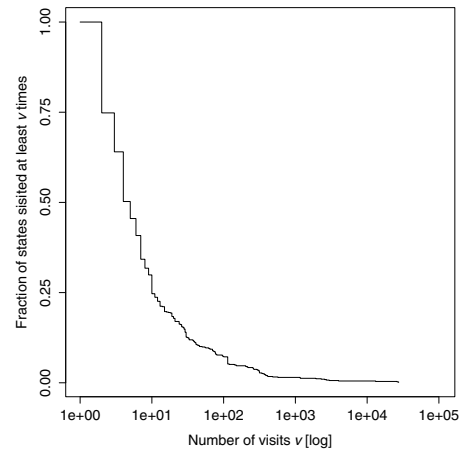


Figure 8: Distribution of the fraction of states visited at least  $v$  times. The fraction is computed with respect to the total number of states visited in 200 runs of 130000 timesteps each. Sequence-recognition test case.

ing graphs of the observed state transitions, we observe that the limited region of state space utilized is again organized in sets of clusters of states. The graph derived in a typical case of sequence recognition can be found as supplementary material (Garattoni et al., 2013). At the top of the graph, a set of nodes allows the robot to navigate on the background with its LEDs off until the first colored stripe is found. Then, two clusters of nodes are responsible of the next action, depending on the detected color (turn LEDs on if black, turn LEDs off if gray). Once the first color has been recognized, the BN goes into a new region, dual to the first. Here, we find another area for the background color and two clusters of nodes for the black and gray with actions swapped with respect to the first region. When also the second color is recognized, the dynamics return back to the first area, reusing the same states to recognize a sequence of any length. This analysis shows that the memory, in our case the last color recognized, is stored in the state space in which the BN operates.

Similarly to the corridor-navigation test case, each cluster of states is devoted to the execution of a particular functional behavioral unit of the robot. To support the observation and show that each cluster unfolds around few topical states and a series of other nodes gradually less important, we report in a plot the distribution of the fraction of states visited at least  $v$  times. The results for a typical case of sequence recognition are depicted in Figure 8.

The properties discussed so far allow the employment of the same heuristic used for the first test case to obtain a compact FSM representation of networks performing sequence recognition. From the graph described, we derived the FSM shown in Figure 9.

From the comparison of the FSM in Figure 9 and the one

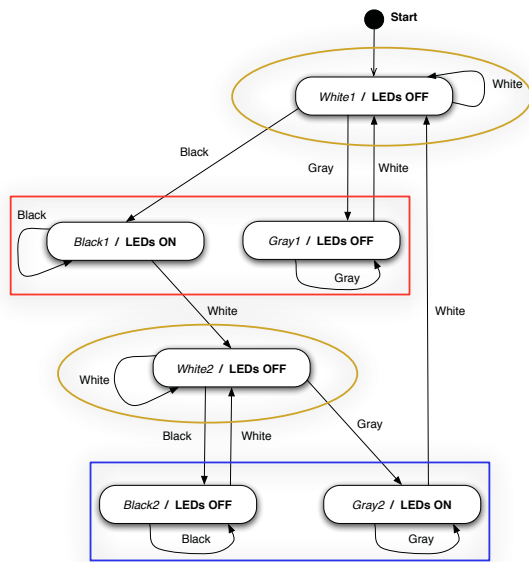


Figure 9: Finite state machine of the state space graph. Sequence recognition.

in Figure 7, it is possible to notice the influence of the different nature of the two tasks. The interesting aspect to highlight is the representation of the memory required by the sequence-recognition test case. In the corridor-navigation FSM, the action executed by the robot at each instant in time is determined only by the current observation of the world. This is due to the fact that corridor navigation is a memory-less task in which the robot is not required to keep memory of the past but it can simply react to the current stimuli of the environment. On the contrary, the FSM performing sequence recognition can activate different actions, e.g. LEDs on or LEDs off, for the same observation, e.g. the detection of the black color, depending on the previous state. Therefore, the memory that the robot needs to keep track of the last recognized color is stored in the phase space in which the BN operates. More precisely, the memory of the past is represented by the area of state space utilized by the network at a certain time, which is function of all the previous robot-environment interactions.

## Conclusion

In this paper, we have exploited the properties of the automatic design of BN-robot control software to synthesize FSM representations of the robot program. This result has been made possible by an analysis performed on the state space of the best networks obtained at the end of the design process. In particular, the exploration revealed two crucial properties: (i) the trajectories of the BNs controlling the robots are confined in very small areas of the state space and (ii) the dynamics are organized in clusters of states occupy-

ing different areas of the state space, each corresponding to a different set of actions to perform. These results allowed us to outline a procedure to derive a compact view of the best performing network behaviors in terms of FSMs.

A major advantage of this method over current automatic design of FSM controllers for robots is that it does not require any assumption on the number of states nor conditions on the transitions between states. This implies that the behavior of the robot is automatically segmented, i.e., the actions composing the robot's behavior do not need to be specified *a priori*. We would like to emphasize that the use of BNs makes it possible to exploit the properties of both NNs and high level representations like FSMs. In fact, most NN-based robot programs define a mapping between sensor readings and actions on the actuators and thus operate as low-level, fine grained programs which are particularly effective in reactive systems. Conversely, FSM control software is usually based on high-level actions and it is suitable for modular control programs which can be also formally verified. With BN robot programs we can combine both characteristics, as BNs can indeed operate low-level and, at the same time, enable the designer to manipulate a FSM description of the robot control software.

The work carried out for this paper is only a first necessary step towards the application of the proposed approach to more complex and demanding tasks. Future work will focus on improving the performance of the design process and defining an automatic method for synthesizing FSMs starting from Boolean networks. These steps are required for a fair comparison of the proposed approach with existing and well refined design methods.

Of course, the approach has also some limitations. First of all, it requires to deal with Boolean inputs and outputs, which could be sometimes problematic. In addition, the FSM is derived by collecting samples of BN trajectories and a trade-off between precision and computational complexity has to be found.

## Acknowledgements

This work was partially supported by the EU project ASCENS (grant 257414). Mauro Birattari acknowledges support from the Belgian F.R.S.-FNRS, of which he is a Research Associate.

## References

- Ansaloni, L., Villani, M., and Serra, R. (2009). Dynamical critical systems for information processing: a preliminary study. In Villani, M. and Cagnoni, S., editors, *Proceedings of the Satellite Workshops of the International Conference of the Italian Association for Artificial Intelligence (AIIA09)*, pages 210–218. Reggio-Emilia, Italy.

- Bar-Yam, Y. (1997). *Dynamics of complex systems. Studies in nonlinearity*. Addison-Wesley, Reading, MA.
- Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3(4):471–511.
- Dorigo, M. (1994). Learning by probabilistic boolean networks. In Ruck, D., Wada, M., and Bounds, D., editors, *1994 IEEE International Conference on Neural Networks: IEEE World Congress on Computational Intelligence*, pages 887–891. IEEE Press, Piscataway, NJ.
- Fogel, D. B. (1993). Evolving behaviors in the iterated prisoner’s dilemma. *Evolutionary Computation*, 1(1):77–97.
- Fogel, L. J. (1962). Autonomous automata. *Industrial Research Magazine*, 4(2):14–19.
- Fretter, C. and Drossel, B. (2008). Response of boolean networks to perturbations. *The European Physical Journal B - Condensed Matter and Complex Systems*, 62(3):365–371.
- Garattoni, L., Roli, A., Amaducci, M., Pinciroli, C., and Birattari, M. (2013). Additional material to the paper “Boolean network robotics as an intermediate step in the synthesis of finite state machines for robot control”. Available as <http://iridia.ulb.ac.be/supp/IridiaSupp2013-004/>.
- Gershenson, C. (2004). Introduction to random boolean networks. In Bedau, M., Husbands, P., Hutton, T., Kumar, S., and Suzuki, H., editors, *Workshop and Tutorial Proceedings, Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX)*, pages 160–173. MIT Press, Boston, MA.
- Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467.
- Kauffman, S. A. (1991). Antichaos and Adaptation. *Scientific American*, 265:78–84.
- Koza, J., Keane, M., and Streeter, M. (2003). Genetic programming’s human-competitive results. *IEEE Intelligent Systems*, pages 25–31.
- Lipson, H. (2005). Evolutionary robotics and open-ended design automation. In Cohen, B., editor, *Biomimetics*, pages 129–155. CRC Press.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotcz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In Gonçalves, P., Torres, P., and Alves, C., editors, *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65. IPCB, Castelo Branco, Portugal.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary robotics*. The MIT Press.
- O’Neill, M., Vanneschi, L., Gustafson, S., and Banzhaf, W. (2010). Open issues in genetic programming. *Genetic Programming and Evolvable Machines*, 11(3-4):339–363.
- Patarnello, S. and Carnevali, P. (1986). Learning networks of neurons with boolean logic. *Europhysics Letters*, 4(4):503–508.
- Petrovic, P. (2007). Strengths and weaknesses of FSA representation. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation, GECCO ’07*, pages 723–725. ACM, New York, NY, USA.
- Pinciroli, C., Trianni, V., O’Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Birattari, M., Gambardella, L. M., and Dorigo, M. (2012). ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295.
- Ribeiro, A. S., Kauffman, S. A., Lloyd-Price, J., Samuelsson, B., and Socolar, J. E. S. (2008). Mutual information in random boolean models of regulatory networks. *Physical Review E*, 77(1):011901.
- Roli, A., Manfroni, M., Pinciroli, C., and Birattari, M. (2011). On the design of boolean network robots. In *Proceedings of EVOApplications 2011*, Lecture Notes in Computer Science, pages 43–52. Springer, Berlin, Germany.
- Serra, R., Villani, M., Graudenzi, A., and Kauffman, S. A. (2007). Why a simple model of genetic regulatory networks describes the distribution of avalanches in gene expression data. *Journal of Theoretical Biology*, 246(3):449–460.
- Serra, R. and Zanarini, G. (1990). *Complex Systems and Cognitive Processes*. Springer-Verlag, Secaucus, NJ.
- Sun, R. and Giles, L. C. (2001). Sequence learning: From recognition and prediction to sequential decision making. *IEEE Intelligent Systems*, 16(4):67–70.
- Yamauchi, B. (1993). Dynamical neural networks for mobile robot control. Technical report, NRL Memorandum Report AIC-033-93 (Naval Research Laboratory).
- Yamauchi, B. and Beer, R. D. (1994). Sequential behavior and learning in evolved dynamical neural networks. *Adaptive Behavior*, 2(3):219–246.



# Predator prey molecular landscapes

Adrien Padirac<sup>1</sup>, Alexandre Baccouche<sup>1</sup>, Fujii Teruo<sup>1</sup>, Andre Estevez-Torres<sup>2</sup> and Yannick Rondelez<sup>1</sup>

<sup>1</sup>LIMMS/CNRS-IIS, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan

<sup>2</sup>LPN, CNRS, route de Nozay, 91460 Marcoussis, France  
rondelez@iis.u-tokyo.ac.jp

## Abstract

This paper describes the use of molecular programming techniques to build synthetic *in vitro* and spatially distributed reactions networks with tailored topologies. The basic workflow is to use synthetic DNA strands to encode the topologies of molecular interactions of the reaction network. The actual dynamic of the system is provided by enzymatic reactions controlled and templated by these DNA strands. Here we focus on the implementation of a molecular predator-prey ecosystem. We thus create two autocatalytic DNA amplifications reactions and connect them through predation – the second DNA-species consumes the first one to fuel its growth. We also ensure that these species have a limited lifetime in the test tube. We are therefore able to detect sustained oscillations of the two molecular species, as predicted and observed for real ecosystems. This is the first time that predator prey oscillations are observed in a chemical system. We further expand the analogy between chemical and animal networks by introducing additional interactions such as symbiosis, the mutually beneficial interaction between two species. Interestingly, competition also arises quite naturally from the physical substrate that is used in the modeling process and displays remarkable dynamic consequences such as synchronization or chaos. Finally we report the construction of spatially distributed chemical ecosystems, and the observation of their spatiotemporal behaviors, in particular traveling and spirals dual waves of molecular hunts.

## Introduction

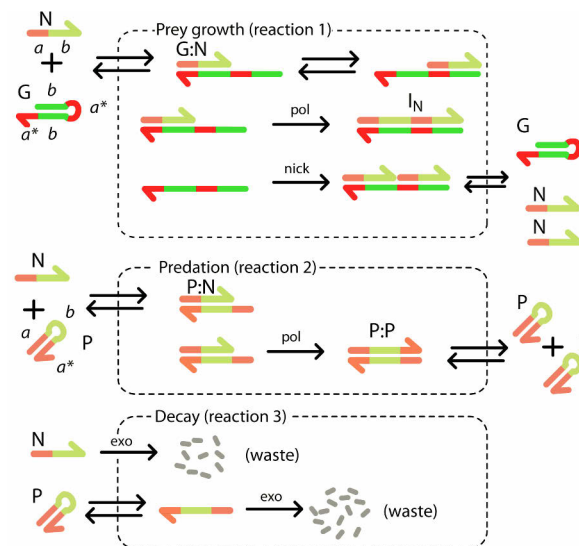
Molecular programming techniques based on synthetic DNA are currently opening unprecedented opportunities for the exploration of molecular informational systems. Because DNA allows easy encoding of molecular interactions and possesses a rich biochemistry, it is possible to reproduce, in test tubes, some of the most fundamental dynamic motifs of biological regulation circuits, like oscillators, bistable switches, etc. (Montagne et al. 2011; Kim & Winfree 2011). This synthetic approach provides a unique opportunity to i) better understand the structure/function relationships at the level of biological circuits; ii) use such molecular devices (computers, controllers, memories, filters ...) into informational chemical systems; and iii) design artificial molecular systems integrating more and more life-like features.

We recently went one step further by demonstrating that molecular programmers need not restrict their inspiration to cellular circuits. Other networks, such as those formed by interdependent species (ecosystems) can also be reproduced

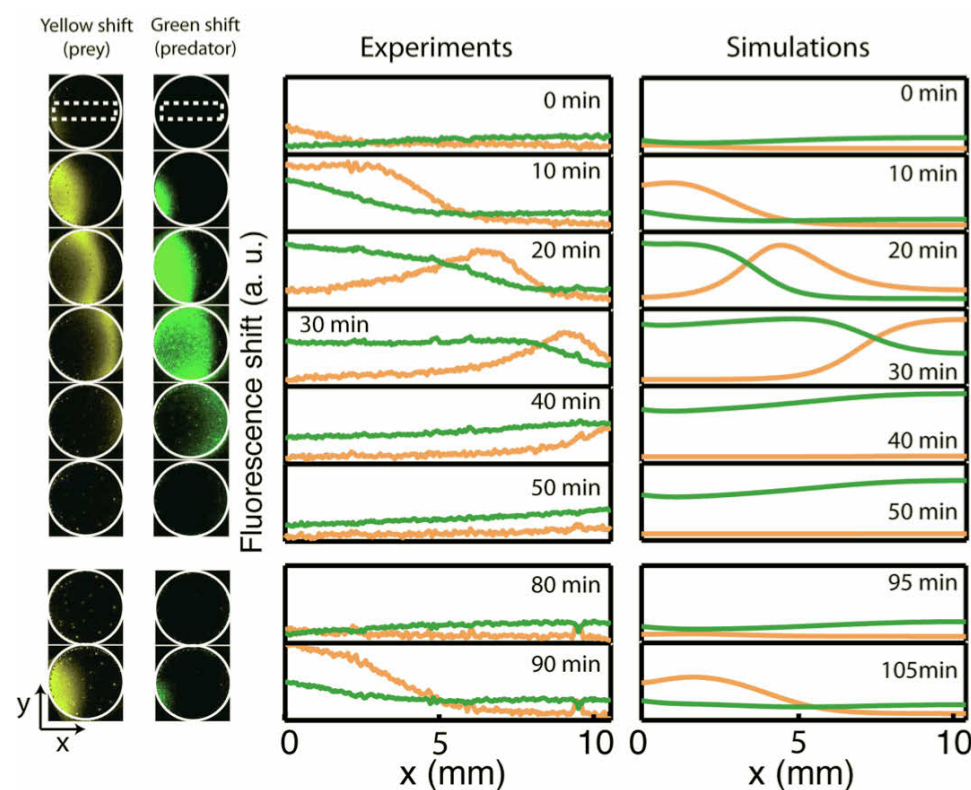
using molecular tools. Our demonstration is based on the predator-prey example, the basic motif of many ecosystems. This particular motif is well known because, somewhat disconcertingly, the simple interaction between a prey and its predator typically leads to sustained oscillations of both populations (Lotka 1920).

## Results and discussion

We have encoded the topology of PP interactions in a DNA-based molecular program. The information concerning the topology of the network is genetically stored in a 20-base-long ssDNA, G (for grass), which direct the growth of complementary preys (N) as follows: N hybridizes to the 3' end of G, to form the partial duplex G:N. This duplex is extended by a polymerase and subsequently nicked by a specific nicking enzyme yielding, upon de-hybridization, two copies of N and an intact template G.



**Figure 1:** Molecular predator-prey network. N, P and G denote the prey, the predator and the template respectively. Harpoon-ended arrows denote DNA strands. Double-sided arrows correspond to DNA hybridization/dehybridization reactions, whereas single-sided arrows indicate irreversible enzymatic transformations. Complementary DNA domains have the same colour. Strands have different hues, light and dark, indicating if they can or cannot be degraded by the exonuclease, respectively. Pol, nick and exo stand for bst DNA polymerase, Nb.BsmI nicking enzyme and ttRecJ exonuclease.



**Figure 2:** Waves of prey (yellow) and predators (green) in a 2D predator-prey molecular experiment. Left: Time-lapse images of the fluorescent shift in the corresponding fluorescent channels, taken every 10 min in a circular reactor 11 mm in diameter and 200  $\mu\text{m}$  thick. The borders of the reactor are highlighted in white. Middle: profiles along  $x$  of the yellow (prey) and green (predator) fluorescent shifts. Right: 1D reaction-diffusion simulations of the normalized prey and predator concentrations.

Predator P is a 14 bases long palindromic ssDNA. During predation N hybridizes over P and the polymerase extends this adduct to form a double strand P:P. Upon de-hybridization, P:P yields two copies of P. Both active species N and P are degraded to unreactive dNMPs by a 5'→3', processive, ssDNA-specific exonuclease. G is not digested because it bears three protective phosphorothioate modifications in 5'.

We have then confirmed the accuracy of our experimental model by observing sustained chemical cycles in a test tube maintained at a constant temperature. The period is from one to several hours and the oscillations of the two species can be monitored in two fluorescent channels. Tens of cycles can be obtained even in the absence of any exchange of matter. This demonstrates the transposition of an agent-based non-trivial network (and its dynamic behavior) at the molecular scale.

We have further extended the approach by adding additional ecologically-relevant interactions into the molecular ecosystem: competition for shared molecular resources, such as enzymatic catalysts can lead to complex, possibly chaotic behaviors, while symbiosis at the prey level tend to stabilize the steady coexistence of the species (Fujii & Rondelez 2013). We have also integrated a spatial component into the system by moving from well-mixed to reaction-diffusion systems. This has allowed the first observation of synthetic predator-

prey “waves of pursuit and evasion” (Murray 2004) under the microscope (Padirac et al. 2013).

## References

- Fujii, T. & Rondelez, Y., 2013. Predator-prey molecular ecosystems. *Acs Nano*, 7(1), pp.27–34.
- Padirac et al., 2013. Spatial waves in synthetic biochemical networks, *Journal of the American Chemical Society*, DOI: 10.1021/ja403584p.
- Kim, J. & Winfree, E., 2011. Synthetic in vitro transcriptional oscillators. *Molecular Systems Biology*, 7(1).
- Lotka, A.J., 1920. Analytical note on certain rhythmic relations in organic systems. *Proceedings Of The National Academy Of Sciences Of The United States Of America*, 6(7), p.410.
- Montagne, K. et al., 2011. Programming an in vitro DNA oscillator using a molecular networking strategy. *Molecular Systems Biology*, 7(1).
- Murray, J.D., 2004. *Mathematical Biology I*, Springer Verlag.

# The role of backward reactions in a stochastic model of catalytic reaction networks

Alessandro Filisetti<sup>1,4</sup>, Alex Graudenzi<sup>2</sup>, Chiara Damiani<sup>2</sup>, Marco Villani<sup>1,3</sup> and Roberto Serra<sup>1,3</sup>

<sup>1</sup>European Centre for Living Technology, University Ca' Foscari of Venice, Italy

<sup>2</sup>Dept. of Informatics, Systems and Communication, University of Milan Bicocca, Italy

<sup>3</sup>Dept. of Physics, Informatics and Mathematics, Modena and Reggio Emilia University

<sup>4</sup>CIRI, Energy and Environment Interdepartmental Center for Industrial Research, University of Bologna

corresponding author: alex.graudenzi@unimib.it

## Abstract

We investigate the role of backward reactions in a stochastic model of catalytic reaction network, with specific regard to the influence on the emergence of autocatalytic sets (ACSs), which are supposed to be one of the pre-requisites in the transition between non-living to living matter.

In particular, we analyse the impact that a variation in the kinetic rates of forward and backward reactions may have on the overall dynamics.

Significant effects are indeed observed, provided that the intensity of backward reactions is sufficiently high. In spite of an invariant activity of the system in terms of production of new species, as backward reactions are intensified, the emergence of ACSs becomes more likely and an increase in their number, as well as in the proportion of species belonging to them, is observed. Furthermore, ACSs appear to be more robust to fluctuations than in the usual settings with no backward reaction.

This outcome may rely not only on the higher average connectivity of the reaction graph, but also on the distinguishing property of backward reactions of recreating the substrates of the corresponding forward reactions.

## Introduction

Models of catalytic reaction networks have been widely investigated in the last decades, with different goals and purposes, yet mostly in regard to the broad theme of the origin of life and with the design of artificial *protocells* (Carletti et al., 2008; Filisetti et al., 2010; Rasmussen et al., 2004; Serra et al., 2007; Szostak et al., 2001).

In particular, in the quest for a reasonable theory describing the transition from non-living to living matter, many frameworks have been proposed, among others the *metabolic-first* scenario (Dyson, 1985; Smith and Morowitz, 2004; Wächtershäuser, 1990; de Duve, 1982), the *protein-first* hypothesis (Oparin, 1924; Fox, 1974; Lee et al., 1996, 1997; Issac and Chmielewski, 2002), the *compartmentalization* (Bachmann et al., 1992), the *compositional* approach (Segré et al., 1999; Segre et al., 1998; Segré and Lancet, 2000) and the *gene-first* hypothesis included in the RNA world theory (Gilbert, 1986; Müller, 2006; De Luca et al., 2007; Anastasi et al., 2007; Talini et al.,

2009; Rios and Tor, 2009; Budin and Szostak, 2010). Even if the dispute is far from being concluded (Cornish-Bowden and Cárdenas, 2008; Stano and Luisi, 2010; Schrum et al., 2010; Budin and Szostak, 2010), one of the underlying key requirements in most of these theories is that the production of the molecular species involved in the transition relies on robust reaction pathways.

In this regard, some theories account for linear chemical pathways capable of producing the sufficient amount of species at energy-rich sites, e.g. hydrothermal vents (Ogasawara et al., 2000) or under plausible prebiotic conditions (Costanzo et al., 2009). Nevertheless, in most of the cases the emergence of sets of collectively self-replicating molecules, i.e. autocatalytic cycles (or *autocatalytic sets*, ACSs from now on)<sup>1</sup> appears to be an essential requirement to achieve the self-sustenance and the evolvability of the system. Indeed, there are many examples of ACSs in current biological systems, which are the outcome of billions of years of evolution. Therefore, the investigation of the generic properties of catalytic reaction networks, with particular respect to the sufficient conditions for the emergence of ACSs and the characterisation of their dynamical properties, is fundamental<sup>2</sup>.

<sup>1</sup>A classical definition of ACS is that of a subset of chemicals in which the production of each element is catalysed by at least another elements belonging to the subset (Kauffman, 1986). Hereinafter, a more formal definition with specific regard to our model will be provided.

<sup>2</sup>It is very important to remark that the presence of ACSs only is not sufficient to define *life*, which it is largely believed to require also the presence of a container that separates the living system from the environment, as well as a coupling between the replication rate of the internal molecules and the growth and division rate of the container. This theme is at the centre of the research on protocells.

In previous works (Serra et al., 2007; Filisetti et al., 2008; Carletti et al., 2008) we proved that, once that such a coupling is achieved, the rates of the replication of the internal molecules and that of the growth of the container tend to spontaneously synchronise through successive divisions. This also leads to an exponential growth of the population of protocells that, in turns, implies a Darwinian selection process among them (Munteanu et al., 2006).

Furthermore, in Serra et al. (2013) we introduced the first known

To this end, different models have been proposed by, e.g., Dyson (Dyson, 1985), Eigen and Schuster (Eigen and Schuster, 1977; Eigen and McCaskill, 1988; Eigen and Schuster, 1978), Kauffman (Kauffman, 1986; Hordijk et al., 2010; Hordijk and Steel, 2004; Mossel and Steel, 2005), Jain and Khrihna (Jain and Krishna, 1998), Lancet (Segre et al., 1998; Segré and Lancet, 2000), Kaneko (Kaneko, 2006) and Vasas and Fernando (Vasas et al., 2012). Despite the theoretical differences, most of the models predict a phase transition leading to the spontaneous emergences of ACSs, after matching certain key conditions, either structural or dynamical according to the cases. Note that, given that the presence of ACSs may eventually lead the system to display remarkable discrepancies between the concentration of the involved molecules from that of analogous systems with no ACSs, this could be actually investigated through wet-lab experiments.

Nonetheless, it is indeed difficult to detect the emergence of ACSs in lab experiments and this could be due, on the one hand, to the somehow drastic simplifications at the basis of the theoretical models or, on the other hand, to the fact that real experiments never matched the requirements suggested by the theories.

In order to fill the gap between theories and experiments and provide insights for further experimentation, in Filisetti et al. (2011c) we introduced a novel model of catalytic reaction network, based on a fully stochastic framework and in which the system complexity can grow according to the dynamics, through the creation of new species and reactions. The model takes inspiration from the works by Kauffman (Kauffman, 1986) and its subsequent developments by others (Bagley et al., 1989; Bagley and Farmer, 1992; Farmer and Kauffman, 1986; J.D.Farmer et al., 1986).

The model considers abstracted entities accounting for monomers and polymers (i.e. *species*) and simplified interactions among them, in terms of cleavages (i.e. the cutting of two species) and condensations (i.e. the concatenation of two species). The key constraint of the model is that each reaction must be catalysed in order to occur. In this regard, any species in the system can be selected to be the catalyst of any possible reaction with a certain probability. The system's dynamics is then stochastically simulated within an open flow reactor. By using a stochastic framework, it is possible to consider in an adequate way the relevance of noise, random fluctuations and low-numbers-effects on the overall dynamics, most of all when dealing with systems

---

stochastic model in which a catalytic reaction network is modelled within a simplified model of protocell. Although a stochastic description has been adopted also in Mavelli and Ruiz-Mirazo (2013), our model of protocell deals with the capability to create new molecular species by means of the reactions present in the system.

close to the phase transition in which the emergence of ACSs becomes plausible.

We remark that the focus of the model is *not* on the detailed characterisation of the entities and reactions of a specific chemical system, but rather on the investigation of the dynamical behaviour that emerges from the interaction of simple entities, with the final goal of deciphering the generic (or universal) properties, that is, those that are shared by a possibly broad range of different chemical systems. In particular, we aim at determining the minimal conditions for the emergence of ACSs and the sensitivity of the phenomenon to variations in some key parameters.

In this regard, in our previous works we studied in depth the influence that variations in some of the key parameters of the system has on the overall dynamical behaviour and on the production of ACSs.

One first result we obtained was to detect that a variation in the composition of the set of molecules present at the beginning of the simulation does not seem to remarkably affect the dynamics of the systems, whereas modifications in the incoming flux seem to deeply influence the overall behaviour (Filisetti et al., 2011c).

For this reason, we focused our attention on the incoming flux composition and diversity (Filisetti et al., 2011a). The results of the analysis that we performed showed that the a variation in the number of distinct species belonging to the incoming flux influences the general activity of the system: considering a fixed overall incoming flux concentration, the larger the number of diverse species (regardless their lower individual concentration), the higher the activity of the system, in terms of overall number of species and molecules, yielding a larger number of ACSs. On the contrary, the length of the polymers belonging to the flux seems not to be so relevant.

Another key parameter of the system, the average residence time of the molecules within the reactor, was also analysed, suggesting that the larger the residence time is, the higher the probability of emergence of ACSs is.

In another work, presented at *ECAL 2011* (Filisetti et al., 2011b), we introduced some plausible energy constraints associated to specific types of reactions, to investigate whether and how the introduction of a form of energy could affect the dynamics and the emergence of ACSs. Preliminary analyses showed that there exists an optimal combination of two key parameters, i.e. the incoming flux of energy carriers and the energization kinetic constant (which account for the amount of energy available for endoergonic reactions) and that this combination ensures a larger production of new species. Further research is needed for a better understanding of the phenomenon.

Finally, one of the most important results was to highlight the general fragility of the ACSs that have been observed. In fact, their existence usually depends on same rare molecules



and reactions, which may disappear because of random fluctuations, hence preventing the autocatalytic closure over a significant span of time. This outcome could provide one of the first possible explanations of the difficulty in observing the emergence of ACSs in wet-lab experiments.

In line with this methodological approach, within this work we carry on with the analysis of the key parameters of the system, in order to provide a more complete and coherent picture of the phenomenon.

In particular, we here relax one of the constraints of the classical formulation of the model, that is, the exclusion of backward reactions. In previous studies backward reactions were neglected, by hypothesising that the Gibbs energy  $\Delta G$  for any reaction is large enough to maintain the system far from the chemical equilibrium. Considering that backward reactions *do* occur in nature, here we want to investigate their role in the overall dynamics.

We define a cleavage (respectively, a condensation) as the *backward reaction* of a specific condensation (respectively, cleavage), i.e. the *forward reaction*, if:

- its products are the substrates of that condensation (respectively, cleavage),
- its substrates are the products of that condensation (respectively, cleavage).

The analysis has then been focused on the effects of the relative strength of the rates of the forward and the backward reactions.

In section II the model will be briefly outlined. In section III results of the simulations with backward reactions will be shown. Finally, in section IV the discussion and some indications for future works will be presented.

## The model

A detailed description of the model can be found in (Filisetti et al., 2011a,c), here we will only outline the key features. The model represents an open system in which monomer and polymers, i.e. the *species*, are involved in catalysed reactions. Every species  $x_i, i = 1, 2, \dots, N$  is defined by an ordered string of letters selected from an arbitrary alphabet (e.g. A, B, C...) and its amount, either concentration or quantity, i.e. number of *molecules*. The only allowed reaction types are: i) *cleavage*, the splitting of two species (e.g.  $AAAB \rightarrow A + AAB$ ) and the ii) *condensation*, i.e. the concatenation of two species (e.g.  $BBA + BA \rightarrow BBABA$ ), which requires an intermediate step involving the formation of a temporary complex between the substrate and the catalyst.

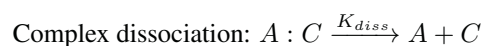
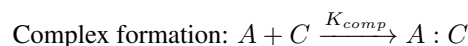
We neglect spontaneous reactions by assuming that there is a sufficiently high activation energy for any reaction scheme. Therefore, only catalysed reactions are allowed and every

species  $x_i$  (longer than a specific threshold) can be selected to be the catalyst of a given reaction with a certain (uniform) probability  $p_i = p, i = 1, 2, \dots, N$ . Therefore, the reaction scheme is defined in a probabilistic way, i.e. in different simulations the same species can be the catalyst of distinct reactions. Besides, the initial reaction scheme can dynamically evolve and increase in dimension because of the creation of new species, which can be (probabilistically) involved in either novel reactions as substrates, products or catalysts, provided that the coherence with the existent reaction scheme is maintained. The set of couples  $\{\textit{species}, \textit{reaction}\}$  in which the species catalyses the reaction defines the *chemistry* of the system, because it describes a coherent possible artificial world. Hence, it is possible to simulate distinct chemistries or to keep the chemistry fixed and simulate different time histories.

In the classical formulation of the model, backward reactions are also excluded, by hypothesising that the Gibbs energy  $\Delta G$  for any reaction is large. The main goal of this work is to investigate the implications of relaxing this constraint.

A possible example of each reaction type is shown:

- Cleavage:  $AB + C \xrightarrow{K_{cleav}} A + B + C$
- Condensation: (whole reaction:  $A + B + C \rightarrow AB + C$ )



A and B are two random species standing for the substrates of the specific reaction, C is the catalyst of that reaction and  $A : C$  is the transient complex.  $K_{cleav}$ ,  $K_{comp}$ ,  $K_{diss}$  and  $K_{cond}$  respectively are the kinetic rates of cleavage, complex formation, complex dissociation and final condensation<sup>3</sup>. The outgoing flux is simulated by assigning a common decay time  $K_{out}$  to each species and complex. The incoming flux rate  $K_{in}$  is measured in moles per second and the average residence time is given by  $1/K_{out}$ .

The dynamics of the system is simulated through the well-known Gillespie algorithm (Gillespie, 1977) for the stochastic simulation of chemical reaction system, with the key modification of allowing the creation of new species and reactions that are not present in the initial conditions. In particular, the system is modelled within a *continuous stirred-tank reactor* (CSTR), which allows continuous

<sup>3</sup>Notice that a parameter sensitivity analysis of the model was presented in Damiani et al. (2013). The main goal was to identify those kinetic parameters that mainly influence the ability of the system to increase the diversity of the species.

ingoing and outgoing fluxes of molecules. In another work (Serra et al., 2013) we introduced a semi-permeable membrane to separate the catalytic reaction network from the environment.

Two possible representations of the system are possible, which results in different graphs. The first concerns the catalytic activity of the system: an edge from  $x_c$  to  $x_i$  is drawn if species  $x_c$  is the catalyst of the reaction in which the species  $x_i$  is one of the products. This leads to the so-called *catalyst-product* graph. The second representation regards the assembling activity: an edge from  $x_j$  to  $x_i$  is drawn when  $x_j$  is a substrate involved in a reaction in which  $x_i$  is one of the products. This allows to draw the *substrate-product* graph.

Besides, the adoption of an asynchronous stochastic framework implies the problem of detecting cycles in a univocal way. We lastly decided to introduce a graph in which every edge (either catalyst-product or product-substrate) is maintained only if the specific reaction occurs within an arbitrary temporal window,  $W$ . We call it the *actual reaction graph* and can be applied to both the catalyst-product and the substrate-product graphs. In this way, the influence of very rare reactions is neglected and cycles can be coherently detected.

In particular, in the context of our model, we define as ACS a subset of species which belong to a *strongly connected component* (SCC) in the catalyst-product actual reaction graph<sup>4</sup>.

**The introduction of backward reactions.** The goal of this work is to investigate whether and how the introduction of backward reactions may influence the overall dynamics of the system and, in particular, with respect to the emergence of ACSs.

To this end, we define a backward reaction for any existing reaction of the system (which will be defined as forward reaction), relative to both cleavages and condensations (see the definition of forward and backward reactions in the previous section). In the example scheme above, the condensation is the backward reaction of the cleavage (or the other way round).

The analysis is then focused on the variation of a key parameter  $R$ , which accounts for the relationship between the forward and the backward reactions kinetic rates, and is defined as follows. Note that, given that in the current configuration of the system we set  $K_{comp} = K_{cond}$  for all the condensations, only  $K_{cond}$  will be included in the definition.

We distinguish two cases:

- if the forward reaction is a cleavage, given any  $K_{cleav}$ :

$$R = \frac{2K_{cleav}}{K_{cond}} \rightarrow K_{cond} = \frac{2K_{cleav}}{R} \quad (1)$$

- if the forward reaction is a condensation, given any  $K_{cond}$ :

$$R = \frac{K_{cond}}{2K_{cleav}} \rightarrow K_{cleav} = \frac{K_{cond}}{2R} \quad (2)$$

Varying  $R$  it is possible to define different ratios between the rates of forward and backward reactions and, accordingly, given the kinetic rates of any forward reaction to determine those of the corresponding backward reaction<sup>5</sup>.

## The simulations

The benchmark for this kind of analysis is the case in which no backward reactions are considered and that will be indicated with *NOREV* from now on. We then considered 4 values of  $R = 1, 10, 100, 1000$ .

We created 10 different chemistries and for each of them we varied the value of  $R$  only (simulating 10 different histories, for a total of 500 distinct simulations), in order to disentangle the effect of its variation on the dynamics. The details of the simulations can be found in the caption of Fig. 1.

In Fig. 1 we display the (average) number of distinct species present in at least one copy in time. No remarkable differences are detectable in the number of different species, which reaches an asymptotic value around 60 after a transient whose length is around 300 seconds, in all the cases. Notice that the number of distinct species (which somehow accounts for the *diversity* of the system) does not depend on the flux dynamics only, but on the general capability of the system of generating new species and reactions. Hence, this outcome suggests that the overall activity of the system, in terms of production of new species, is not enhanced by the introduction of backward reactions.

Moreover, a relatively moderate variation is observed also with regard to the asymptotic total number of molecules in the system, which is around 30.000 for all the distinct cases (not shown here).

The (average) number of molecules and that of species belonging to ACSs are shown in Fig. 2. We here have a indeed remarkable result: in correspondence of the lowest values of  $R$  (i.e. proportionally faster backward reactions) we observe a clear increase of the percentage of both molecules and species belonging to ACS, starting from the

<sup>4</sup>See the footnote number 1 at page 1 for a more general definition

<sup>5</sup>Notice that the factor 2 in Eq. 1 and 2 was chosen, in accordance with our previous works (Fuchslin et al., 2010), to roughly balance the speed of the cleavage, which is a one-step reaction, with that of the condensation, which is a 2-steps reaction.

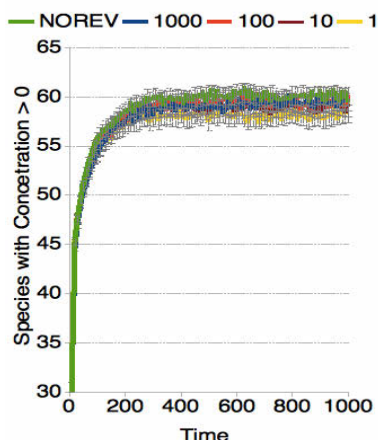


Figure 1: Variation of the average number of species present in the system in time. The five lines represent the case with no backward reactions (*NOREV*) and those of systems characterised by values of  $R = 1, 10, 100$  and  $1000$ . The x axis represents time (in seconds). The values are averaged over 100 different simulations for each value of  $R$ . The bars represent the standard error.

The settings of the simulations are the following.

Alphabet: [A,B]; probability of catalysis,  $p$ : 0.00097; volume of the reactor:  $1e^{-18}L$ ; overall concentration =  $1e^{-4}M$ ; set of species in the influx: all the species up to length 4; minimum polymer length to have catalytic activity: 2; baseline  $K_{cond} = 50M^{-1}sec^{-1}$ , baseline  $K_{comp} = 50M^{-1}sec^{-1}$ , baseline  $K_{cleav} = 25M^{-1}sec^{-1}$ , baseline  $K_{diss} = 1e^{-6}sec^{-1}$ ; influx rate =  $1e^{-21}mol/sec$ ; simulation time: 1000 sec. 10 different chemistries are created, for each chemistry 5 different systems are created: with no backward reactions, with  $R = 1, 10, 100$  and  $1000$ ; for each system 10 different histories are simulated. The number of simulations is so 500.

benchmark case of no backward reactions, in which less than 5% of the molecules and of the species belong to ACSs (and analogously for slower backward reactions, i.e.  $R = 100$  and  $1000$ ), then the case of  $R = 10$ , in which around 20% of the molecules and of the species are in ACSs and up to case in which  $R = 1$ , which involves around 40% of the molecules and 45% of the species in the ACS dynamics. This result hints at a very important consideration: the faster the backward reactions are<sup>6</sup>, the more likely the emergence of ACSs with a large number of molecules and species is. It is even possible to hypothesise a threshold in  $R$  after which the emergence of large ACS becomes very likely, which would be, in this case, between  $R = 10$  and  $R = 100$ .

<sup>6</sup>Meaning proportionally faster with respect to forward reactions.

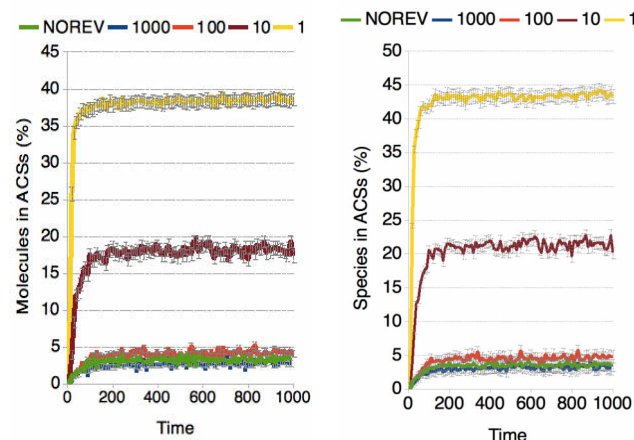


Figure 2: Variation of the average percentage of molecules (left) and species (right) belonging to ACSs in time, with respect to the cases: *NOREV*,  $R = 1, 10, 100$  and  $1000$ . The x axis represents time (in second). The bars display the standard error. The percentage is computed by looking at the molecules and species present at any time step in the system.

In Fig. 3 we report the variation of the number of ACSs (left) and that of percentage of species belonging to ACSs (right) in time, with respect to all the different simulations. Each row of the graph represents a distinct simulation, so it is possible to follow the dynamical evolution of any simulated system, with regard to these two key variables. By looking at the left graphs, regarding the number of ACSs in time, one first important result proves what stated above by analysing the average values. In correspondence of lower values of  $R$  a larger number of simulations is characterised by: i) the emergence of at least one (usually robust in time) ACS, ii) a larger number of distinct ACSs. Whereas for systems with no or very slow backward reactions (e.g.  $R = 1000$ ) in many case no ACSs emerge, when ACSs emerge are often not persistent in time (showing an oscillatory fashion) and the maximum number of observed ACSs is around 4, for low values of  $R$  ( $R = 1$  or  $R = 10$ ) in almost all the simulations at least one ACS is observed and we even observe simulations which yield a indeed large number of ACSs (up to 10 for  $R = 1$ ).  $R = 100$  seems to characterise an intermediate condition, perhaps close to a phase transition in which the emergence of ACSs becomes indeed likely.

Besides, given that the simulations are ordered in sets of 10 different histories for each one of the 10 chemistries, it is possible to notice how some chemistries are actually more efficient in producing ACSs, by looking at the large clearer stripes (i.e. 10 rows, corresponding to 10 histories of the same chemistry), e.g. for  $R = 10$  or  $R = 1$ .

In the right panels it is possible to observe the percentage of species belonging to ACSs in every simulation. For the cases in which only one or a few ACSs emerge (no

backward reactions or high  $R$ ) the fraction of species belonging to ACSs seems to show a strong correlation with the number of ACSs itself. Nevertheless, in the cases in which a larger number of ACSs emerge (low values of  $R$ ) we notice that this correlation is not always preserved and there are, on the one hand, simulations in which a very large percentage of species belong to a relatively low number of (big) ACSs and, on the other hand, others in which a large number of (small) ACSs involve a relatively low proportion of the species. This outcome would suggest that systems with relatively faster backward reactions display more heterogeneous dynamical behaviours.

It is also important to remark that while in the case with no or slow backward reaction at most 20/25% of the species are involved in the ACS dynamics, for low values of  $R$  this percentage increases dramatically, up to more than 70% for some simulation with  $R = 1$ , in which the dynamics of the system is monopolised by ACSs.

In Fig. 4 we show the percentage of molecules in ACSs at the end of the simulation (i.e. time = 1000) as a function of the average connectivity of the catalyst-product actual reaction graph (at time 1000) for all the different cases.

One can first notice that the systems with lower  $R$  correspond to higher average connectivities. This is a somehow expected result, given that the introduction of backward reactions implies an increase in the number of possible reactions and also that the lower the value of  $R$  is, the higher the probability of occurrence of these reactions is, resulting in a actual reaction graph which is increasingly more connected.

Besides, by looking at the line that interpolates the dots relative to the distinct cases (from the case of no backward reaction to the case of  $R = 1$ ), one can detect an apparently super-linear trend, which may underly some non-linear phenomenon, possibly related to the intrinsic nature of backward reactions. In fact, we here remark that the introduction of backward reactions does not simply imply a larger average connectivity for the system, as one of the features of backward reactions is to continuously recreate (as products) the substrates of the corresponding forward reactions. In particular, this action ensures the maintenance of the chains of reactions that guarantee the continuous flow of materials from the system's input toward the ACS structures. It is unlikely that the same reinforcement action of the ACSs' sustaining chains is provided simply because of the doubling of the number of reactions. In order to avoid the collapse, in fact, each ACS has to exactly guarantee the presence of the materials it is consuming: randomly created reactions have scarce chances to reinforce all the needed chemical species, whereas backwards reactions are automatically pointing toward the correct substrates. Given the autocatalytic nature of the ACS, this action is supposed

to guarantee the presence of the needed catalysts<sup>7</sup>.

Therefore, even though further analyses are needed to address this issue, we may suppose that this phenomenon entails important implications on the dynamics and stability of ACSs and, accordingly, to the percentage of molecules belonging to them.

## Conclusions and further developments

In this work we investigated the role of backward reactions in a stochastic model of catalytic reaction network in an open reactor.

The introduction of backward reactions involves significant changes in the overall dynamics, with particular regard to the emergence of ACSs, provided that their speed (hence, frequency of occurrence) is sufficiently high, as established by the kinetic rates and, in particular, by the proportion between the kinetic rates of forward and backward reactions. In other words, the intensity of backward reactions is fundamental to observe remarkable differences in the overall dynamics.

In detail, despite an observed substantial invariance in the number of different species (i.e. the diversity of the system) produced by the dynamics, as long as the relative values of the rates for the forward and backward reaction are decreased (i.e. the intensity of backward reactions is increased), an always higher number of these species is involved in an always larger number of different ACSs, in an increasing number of different simulations.

Besides, when backward reactions gain intensity the ACSs appear to be also more robust to variations and oscillations in time. This could represent a very significant result, mostly in regard with the dynamical fragility of ACSs that was observed in our previous analyses of systems without backward reactions.

It is also possible to hypothesize the presence of a threshold above which the likelihood of emergence of resistant ACSs dramatically increases.

One partial explanation of this general outcome is that backward reactions indeed add new reactions to the chemistry, leading to an increase of the average connectivity of the reaction network, which has been considered one of the key variables in regard to the emergence of ACSs (Filisetti et al., 2011c; Farmer and Kauffman, 1986; Jain and Krishna, 1998).

Nonetheless, the key property of backward reactions of recreating the substrates of the relative forward reactions could be essential in influencing the process of emergence of ACSs, not only because of the increase of the number of possible reactions, but mostly because of their action of reinforcement in favor of the supply chains supporting the

<sup>7</sup>Notice that we are currently designing experiments in order to separate the effects of doubling the number of reactions from those deriving by simple enabling the backward reactions.



existence of ACS structures. Note that this reinforcement action is stronger when the values of the reactions' forward and backward kinetic constants are closer, a circumstance that may have deeply influenced the chemical composition of the first historically functioning ACS structures. Analyses underway are aimed at addressing this issue. Furthermore, the phenomenon of competition for the same catalyst between forward and backward reactions could be another interesting phenomenon to investigate.

In another work (Serra et al., 2013) we introduced a model of catalytic reaction network in protocell, by considering the simplest possible architecture, that is a semi-permeable membrane that selects the species that can enter or exit the protocell.

Among the various results it was shown that protocells display distinct asymptotic behaviours, according to different variables, a property that has never been observed in CSTRs.

Preliminary analyses on the introduction of backward reactions in the protocell model would suggest that even mildly intense backward reactions would lead the system toward a more homogeneous dynamical behaviour. Even if further investigations are ongoing, this outcome would suggest another interesting role of backward reactions in this kind of system, also hinting at possible differences in the hypothesised threshold on the proportion among the kinetic rates.

## Acknowledgements

We deeply thank Stuart Kauffman for many inspirational discussions.

## References

- Anastasi, C., Buchet, F. F., Crowe, M. A., Parkes, A. L., Powner, M. W., Smith, J. M., and Sutherland, J. D. (2007). RNA: prebiotic product, or biotic invention? *Chemistry & biodiversity*, 4(4):721–39.
- Bachmann, P. A., Luisi, P. L., and Lang, J. (1992). Autocatalytic self-replicating micelles as models for prebiotic structures. *Nature*, 357(6373):57–59.
- Bagley, R. and Farmer, J. D. (1992). Spontaneous emergence of a metabolism. *Artificial Life II. Santa Fe Institute Studies in the Sciences of Complexity*, X:93–141.
- Bagley, R. J., Farmer, J. D., Kauffman, S. A., Packard, N. H., Perelson, A. S., and Stadnyk, I. M. (1989). Modeling adaptive biological systems. *Bio Systems*, 23(2-3):113–37; discussion 138.
- Budin, I. and Szostak, J. W. (2010). Expanding Roles for Diverse Physical Phenomena During the Origin of Life. *Annual review of biophysics*, 39:245–63.
- Carletti, T., Serra, R., Villani, M., Poli, I., and Filisetti, A. (2008). Sufficient conditions for emergent synchronization in protocell models. *J Theor Biol*, 254(4):741–751.
- Cornish-Bowden, A. and Cárdenas, M. L. (2008). Self-organization at the origin of life. *Journal of theoretical biology*, 252(3):411–8.
- Costanzo, G., Pino, S., Ciciriello, F., and Di Mauro, E. (2009). Generation of long RNA chains in water. *The Journal of biological chemistry*, 284(48):33206–16.
- Damiani, C., Filisetti, A., Graudenzi, A., and Lecca, P. (2013). Parameter sensitivity analysis of stochastic models: Application to catalytic reaction networks. *Computational biology and chemistry*, 42:5–17.
- de Duve, C. (1982). PEROXISOMES AND RELATED PARTICLES IN HISTORICAL PERSPECTIVE. *Annals of the New York Academy of Sciences*, 386(1 Peroxisomes a):1–4.
- De Lucaezia, D., Anella, F., and Chiarabelli, C. (2007). Question 5: on the chemical reality of the RNA world. *Orig Life Evol Biosph*, 37(4-5):379–385.
- Dyson, F. J. (1985). *Origins of life*. Cambridge: Cambridge University Press.
- Eigen, M. and McCaskill, J. (1988). Molecular Quasi-Specie. *J Phys Chem*, 81:6881–6891.
- Eigen, M. and Schuster, P. (1977). The hypercycle. A principle of natural self-organization. Part A: Emergence of the hypercycle. *Die Naturwissenschaften*, 64(11):541–65.
- Eigen, M. and Schuster, P. (1978). The Hypercycle: a Principle of Natural Self-Organisation, Part B. *Naturwissenschaften*, 65(7):7–41.
- Farmer, J. and Kauffman, S. (1986). Autocatalytic replication of polymers. *Physica D: Nonlinear Phenomena*, 220:50–67.
- Filisetti, A., Graudenzi, A., Serra, R., Villani, M., De Lucaezia, D., Fuchslin, R. M., Kauffman, S. A., Packard, N., and Poli, I. (2011a). A stochastic model of the emergence of autocatalytic cycles. *Journal of Systems Chemistry*, 2(1):2.
- Filisetti, A., Graudenzi, A., Serra, R., Villani, M., De Lucaezia, D., and Poli, I. (2011b). The role of energy in a stochastic model of the emergence of autocatalytic sets. In T. L., M. G., Bersini, H., Bourguine, P., Dorigo, M., and Doursat, R., editors, *Advances in Artificial Life, ECAL 2011 Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*, pages 227–234. MIT Press, Cambridge, MA.
- Filisetti, A., Graudenzi, A., Serra, R., Villani, M., Fuchslin, R. M., Packard, N., Kauffman, S. a., and Poli, I. (2011c). A stochastic model of autocatalytic reaction networks. *Theory in biosciences = Theorie in den Biowissenschaften*, pages 1–9.
- Filisetti, A., Serra, R., Carletti, T., Villani, M., and Poli, I. (2008). Synchronization phenomena in protocell models. *Biophysical Reviews and Letters (BRL)*, 3(1/2):325–342.
- Filisetti, A., Serra, R., Carletti, T., Villani, M., and Poli, I. (2010). Non-linear protocell models: synchronization and chaos. *The European Physical Journal B*, 77(2):249–256.
- Fox, S. W. (1974). The Protein Theory of the Origin of Life. *American Biology Teacher*, 36:161–172.

- Fuchslin, R. M., Filisetti, A., Serra, R., Villani, M., DeLucrezia, D., and Poli, I. (2010). Dynamical Stability of Autocatalytic Sets. In Fellermann, H., Dörr, M., Hanczyc, M. M., Laursen, L. L., Maurer, S., Merkle, D., Monnard, P.-A., Stoy, K., and Rasmussen, S., editors, *Artificial Life XII, Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pages 65–72. The MIT Press.
- Gilbert, W. (1986). Origin of life: The RNA world. *Nature*, 319(6055):618–618.
- Gillespie, D. T. (1977). Exact Stochastic Simulation of Coupled Chemical Reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361.
- Hordijk, W., Hein, J., and Steel, M. (2010). Autocatalytic Sets and the Origin of Life. *Entropy*, 12(7):1733–1742.
- Hordijk, W. and Steel, M. (2004). Detecting autocatalytic, self-sustaining sets in chemical reaction systems. *Journal of theoretical biology*, 227(4):451–461.
- Issac, R. and Chmielewski, J. (2002). Approaching Exponential Growth with a Self-Replicating Peptide. *Journal of the American Chemical Society*, 124(24):6808–6809.
- Jain, S. and Krishna, S. (1998). Autocatalytic set and the growth of complexity in an evolutionary model. *Phys Rev Lett*, 81:5684–5687.
- J.D.Farmer, Packard, N. H., and Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D*, 22(2):187–204.
- Kaneko, K. (2006). *Life: An Introduction to Complex Systems Biology (Understanding Complex Systems)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- Kauffman, S. A. (1986). Autocatalytic sets of proteins. *J Theor Biol*, 119(1):1–24.
- Lee, D. H., Granja, J. R., Martinez, J. A., Severin, K., and Ghadri, M. R. (1996). A self-replicating peptide. *Nature*, 382(6591):525–8.
- Lee, D. H., Severin, K., Yokobayashi, Y., and Ghadiri, M. R. (1997). Emergence of symbiosis in peptide self-replication through a hypercyclic network. *Nature*, 390(6660):591–4.
- Mavelli, F. and Ruiz-Mirazo, K. (2013). Theoretical conditions for the stationary reproduction of model protocells. *Integrative Biology*, pages 324–341.
- Mossel, E. and Steel, M. (2005). Random biochemical networks: the probability of self-sustaining autocatalysis. *Journal of theoretical biology*, 233(3):327–36.
- Müller, U. F. (2006). Re-creating an RNA world. *Cellular and molecular life sciences : CMLS*, 63(11):1278–93.
- Munteanu, A., Attolini, C. S., Rasmussen, S., Ziock, H., and Solé, R. V. (2006). Generic Darwinian selection in protocell assemblies. DOI: SFI-WP 06-09-032, *SFI Working Papers*, Santa Fe Institute.
- Ogasawara, H., Yoshida, A., Imai, E., Honda, H., Hatori, K., and Matsuno, K. (2000). Synthesizing oligomers from monomeric nucleotides in simulated hydrothermal environments. *Origins of life and evolution of the biosphere : the journal of the International Society for the Study of the Origin of Life*, 30(6):519–26.
- Oparin, A. (1924). *The origin of life on the earth*. Oliver and Boyd, 1st ed., p edition.
- Rasmussen, S., Chen, L., Deamer, D., C.Krakauer, D., Packard, N. H., Stadler, P. F., and Bedau, M. A. (2004). Transitions from Nonliving to Living Matter. *Science*, 303, 963–965.
- Rios, A. C. and Tor, Y. (2009). Model systems: how chemical biologists study RNA. *Current opinion in chemical biology*, 13(5-6):660–8.
- Schrum, J. P., Zhu, T. F., and Szostak, J. W. (2010). The Origins of Cellular Life. *Cold Spring Harbor perspectives in biology*, 2(9):a002212.
- Segré, D., Ben-eli, D., Deamer, D. W., and Lancet, D. (1999). The lipid world. *Origins of Life and Evolution of the Biosphere*, 31(1):119–145.
- Segré, D. and Lancet, D. (2000). Composing life. *EMBO reports*, 1(3):217–22.
- Segre, D., Lancet, D., Kedem, O., and Pilpel, Y. (1998). Graded Autocatalysis Replication Domain (GARD): kinetic analysis of self-replication in mutually catalytic sets. *Orig Life Evol Biosph*, 28(4-6):501–514.
- Serra, R., Carletti, T., and Poli, I. (2007). Synchronization phenomena in surface-reaction models of protocells. *Artificial life*, 123(2):123–38.
- Serra, R., Filisetti, A., Villani, M., Damiani, C., Graudenzi, A., and Panini, T. (2013). A stochastic model of catalytic reaction networks in protocells. *Submitted to Artificial Life*.
- Smith, E. and Morowitz, H. J. (2004). Universality in intermediary metabolism. *Proceedings of the National Academy of Sciences of the United States of America*, 101(36):13168–73.
- Stano, P. and Luisi, P. L. (2010). Achievements and open questions in the self-reproduction of vesicles and synthetic minimal cells. *Chemical communications (Cambridge, England)*, 46(21):3639–53.
- Szostak, J. W., Bartel, D. P., and Luisi, P. L. (2001). Synthesizing life. *Nature*, 409(6818):387–390.
- Talini, G., Gallori, E., and Maurel, M.-C. (2009). Natural and unnatural ribozymes: back to the primordial RNA world. *Research in microbiology*, 160(7):457–65.
- Vasas, V., Fernando, C., Santos, M., Kauffman, S., and Szathmari, E. (2012). Evolution before genes. *Biology Direct*, 7(1):1.
- Wächtershäuser, G. (1990). Evolution of the first metabolic cycles. *Proceedings of the National Academy of Sciences of the United States of America*, 87(1):200–4.

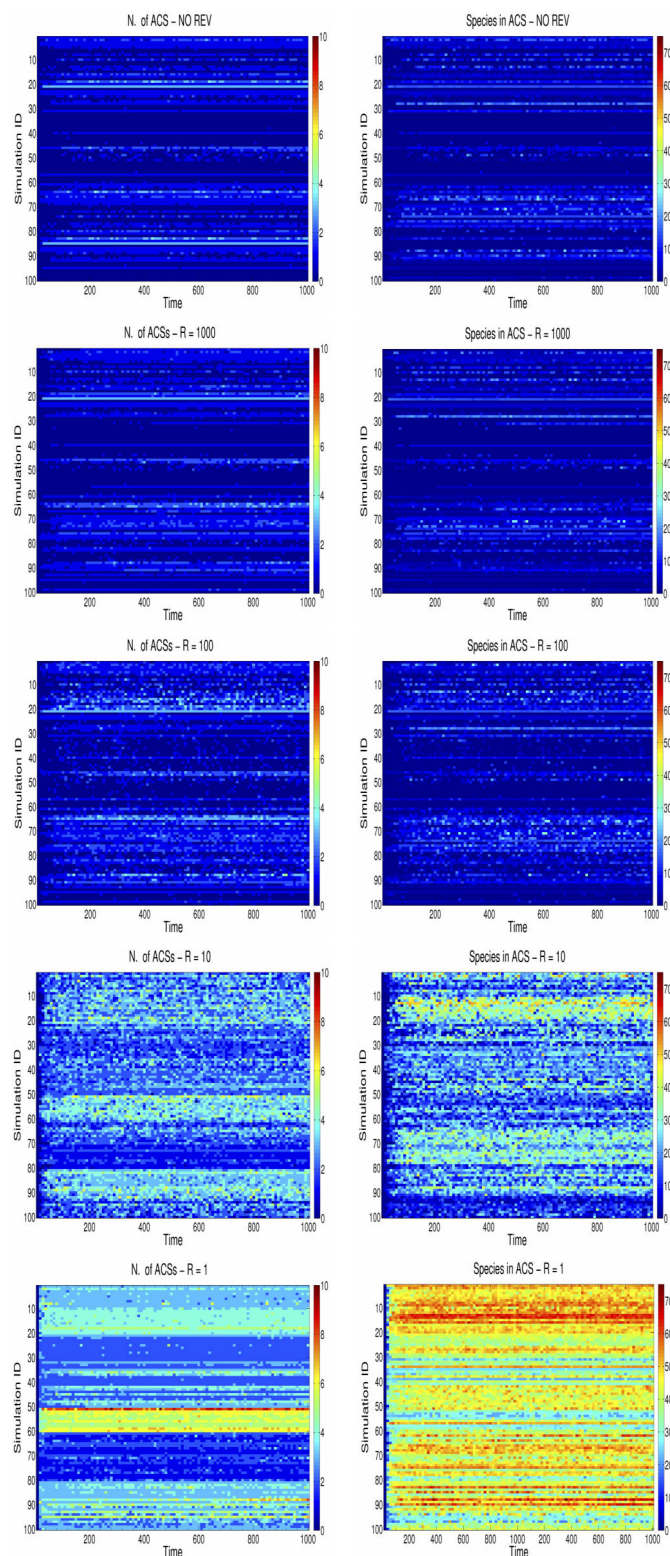


Figure 3: Variation of the number of ACSs (left) and of the percentage of species (on the total) belonging to ACSs (right) for all the simulations, with respect to the cases (in order from top to bottom): *NOREV*,  $R = 1000$ ,  $R = 100$ ,  $R = 10$  and  $R = 1$ . Each row represents one distinct simulation, the x axis represents time. The colours stand for the value of that variable, as in the corresponding colour legend.

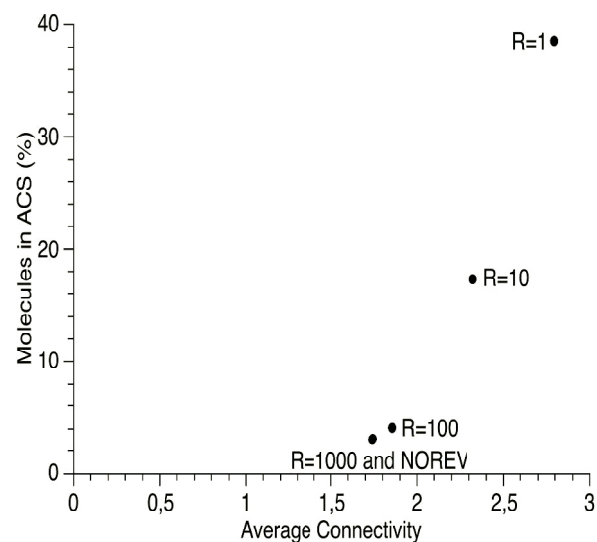


Figure 4: Phase diagram of the relation between the average connectivity of the system and the average percentage of molecules belonging to ACSs at the end of the simulation (i.e. time = 1000 seconds), with respect to the cases: *NOREV*,  $R = 1$ ,  $R = 10$ ,  $R = 100$  and  $R = 1000$ . The x axis stand for the average connectivity and the y axis for the percentage of molecules in ACSs. The colors represent the different cases.



## A cardiopulmonary system for a virtual patient

Vincent Ducharme<sup>1</sup>, Richard Egli<sup>1</sup> and Sylvie Jetté<sup>2</sup>

<sup>1</sup>Department of Computer Science

<sup>2</sup>School of Nursing

Université de Sherbrooke, Sherbrooke, Québec, Canada

vincent.ducharme2@usherbrooke.ca

### Abstract

We present a simulation of a cardiopulmonary system. The simulation is used within a serious game to help for nurses education. It runs in real-time and can be easily modified to represent different illnesses. The system can adequately react when a nurse executes an unexpected action on the patient. The simulator uses a bottom-up design to model the cardiopulmonary system, using simple mathematical models and basic interactions to reproduce high-level and complex behaviors.

### Introduction

Simulation is a good way to learn and practice in a safe environment. The simulation provides useful feedback to help students and trainees to learn from their mistakes. However in healthcare, most of the simulations and simulators require actors or mannequins (Issenberg et al., 2001; Morgan et al., 2006) to practice on. Since these simulations are expensive to use and can not be installed everywhere, it is hard for a student to practice anywhere else than schools and hospitals. Furthermore, most of these existing simulations for nursing education require the supervision of a technician or the use of a lot of parameters, which are often difficult to handle for an inexperienced user. Virtual simulations for nurses' training already exist (Hansen, 2008; Zary et al., 2006). However, most of them lack realism or tools needed by a teacher to provide useful and complete simulation for nursing students. To solve these major inconveniences, we propose a serious computer game coupled with a simulator that relies on interactions of simple components to reproduce complex behaviors. The model of the virtual patient is simplified and based on biological and physiological behaviors. It only specifies atomic parts of the complex system and the basic interactions between them. From these interactions, the required complex behavior emerges and can be studied. This innovative approach will help nurses taking charge of poly-trauma patients at the hospital. Given the low frequency of certain clinical situations in critical care, the use of computer simulations to develop and maintain skills is very well-advised. This active and autonomous learning mode, exercised in a virtual world, will facilitate the transfer of skills

in real-life situations. In the context of a shortage of clinical placement for nursing students, the computer simulation becomes a valuable tool within the reach of educational and health institutions to improve healthcare quality and patient safety.

The human body is a very complex system. Reproducing a perfectly accurate simulation of the human body would require a huge amount of computational resources and a perfect understanding of the underlying physiological processes. This is therefore an uneasy task if not, an impossible one. However, there are many efforts made to construct standards (Coveney et al., 2011; Clapworthy et al., 2008) and common parts (Ellaway et al., 2008) that could be used for a unified model of the human body. More realistic approaches, based on mathematical models like HumMod (Hester et al., 2011), are also developed. The mathematical approach of HumMod contains many variables and use complex formulas that represent the final behavior of the entire system. These models are very precise and require a good understanding of the underlying physiological processes. Researches in physiology and bioengineering are currently conducted to find these mathematical representations. One of the underlying objective of the presented simulator is to reproduce high level behavior without explicitly defining all possible interactions in the system with high level formulas used by these more classical mathematical models.

Most of the models developed for human body simulation use physiological and physical approaches to obtain adequate simulation (Attinger and Anné, 1966; MacIntyre, 2004). Some of them are slow to compute results, mainly due to the complexity of the formulas they used. Since the simulations must execute in real-time within a game engine, these models can not be used. The proposed model relies on such physiological and physical concepts. However, instead of representing complex interactions and using time-consuming computations, the system only uses basic physics formulas for on localized components, making it faster to compute.

In this paper, we present the cardiopulmonary system de-



veloped for the serious game used during the training of nurses. The next section describes the simulator and each of its subsystems. The third section shows some results of the simulator and a discussion about the simulator is presented in the fourth section.

## Simulator description

Our current version of the simulator models the cardiovascular and the respiratory systems. Each system is modeled after its biological functions. The simulator runs in real-time and can thus be linked to a game engine in order to simulate an emergency room with an injured patient. Each of the modeled systems is a simplification of the reality but its behavior is consistent with the physiological response of its human counterpart.

The update process of the simulator relies on the game engine. The game engine updates the simulator periodically using a time step ( $\Delta t$ ). Each system is updated accordingly to that time step by the simulator. For each update of a system, all its sub-components are also updated using this time step.

## Body definition

The simulator relies on a XML file to describe the human body. The entire body description is decomposed into different systems, i.e. cardiovascular, respiratory, nervous, muscular, etc. This paper emphasizes only the description and the simulation of the cardiovascular and the respiratory system. Each system is viewed as a list of connectibles and a list of organs. Each connectible represents the media used for information transfer. Connectibles are grouped in subsets representing logical unit of information diffusion. For example, the blood vessels and how they connect to each other in the right arm will be specified as a subset for the cardiovascular system. In the cardiovascular system, the connectible are called blood vessels. In the respiratory system, they are airways and alveoli. Each connectible can be linked to other connectible to create a circuit. Each subset can also be linked to others, creating a more complex circuit for information diffusion. A connectible can be split into sections of equivalent volume. Each of these sections contain a part of the body fluid that moves into the connectible. For the circulatory system, it is a blood part. Each fluid part contains different metabolites (see the metabolites section for the definition). Fig. 1 illustrates the different compounds of a system for the circulatory system. The use of XML file to specify the different values used by the model have many advantages. Among others, it can be easily modified and it is simple to understand. Since the simulator is used for nursing education, the XML specification is also an easy mechanism to specify injuries to the patient and to create new scenario and cases to practice on.

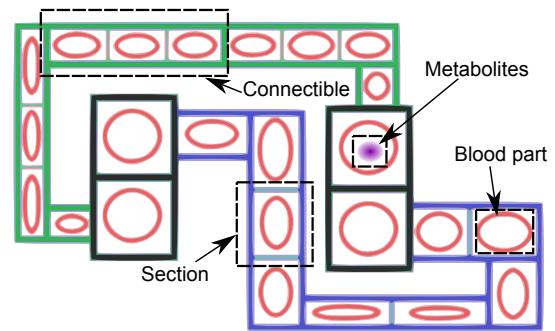


Figure 1: Simplification of the circulatory system to illustrate its different compounds. Each main color represents different subsets of connectible. The outermost bold rectangles are blood vessels (connectible). Inner rectangles are sections. Blood part (red circle) can contains different metabolites (purple circle). Blood vessels are linked together.

## Circulatory system

One of the main system of a human body is the cardiovascular system. The blood flows through the body, diffuses nutrients to various organs and retrieves waste produced by cells. Most of the non-nervous signals of the body use the cardiovascular system to reach their area of action. In the simulator, a virtual bloodstream is used as a transporter and is composed of two circulation loops. The first one is the pulmonary loop. The blood flows from the right ventricle of the heart to the lungs and returns back into the left atrium. The second is the systemic loop. The blood flows from the left ventricle of the heart and returns back into the right atrium after passing through the different parts of the body. The blood flows in blood vessels, creating a delay between the emission of the signals (like hormones) and the start of the associated effect. At the beginning of the systemic circulation loop, the blood vessels, called arteries, divide into smaller vessels. They subdivide until reaching the capillaries bed, modeled as a large container of blood to simplify the simulation. In these capillaries, nutrients contained in the blood can diffuse to irrigated organs. The waste produced by the organs are diffused into the blood of the capillaries. The blood then continues its way back into other blood vessels, called veins. The veins merge together on their way back to the heart. These splitting and merging of blood vessels mix the content of the blood to ensure a better repartition of metabolites into all systems of the body. It is also mimicking very well the human circulatory system since blood vessels also split and merge in the same way. Fig. 2 shows the schematic view of the cardiovascular system.

Each blood vessel, as a connectible, is divided into sections containing different blood parts, each of them having some metabolites. When the simulator updates the cardio-

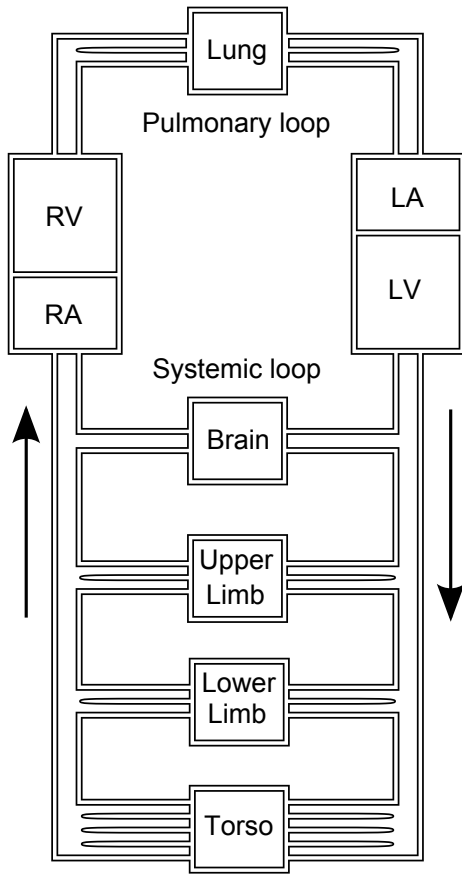


Figure 2: Schematic view of the cardiac model. LA and LV designate respectively Left Atrium and Left Ventricle. RA and RV designate respectively Right Atrium and Right Ventricle. The central boxes of the figure (i.e. Brain, Upper Limb, etc.) represent different connectible subset of the system. The subset contains arteries, veins and capillaries.

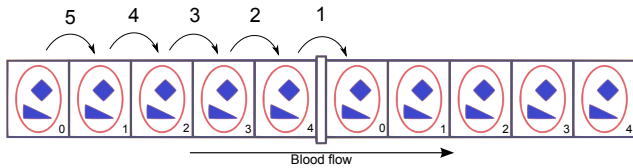


Figure 3: Schematic view of the update in blood vessels. The blood flows from left to right. In this example, each blood vessel contains five sections. Step 1 of the update process moves a certain quantity of blood from the last section of a vessel to the first section of all the next blood vessels, using Eq. 1. Then, step 2 to 5 move a certain volume of each other blood part to the next blood part in the vessel. This circulation of blood is executed in reverse order (from the last section to the first). It avoids transportation of new metabolites through all the blood parts of the blood vessels.

vascular system, each blood part flows through its vessel following the pressure gradient between that part and the next one in the vessel. In the simulation, each vessel is represented by a length ( $L$ ) and a radius ( $r$ ), thus modeled as a finite cylinder. This implies that each section of a blood vessel is also modeled as a cylinder of the same radius as the blood vessel but with a length of  $L/n$  for a vessel of  $n$  sections. To reproduce the pulsative flow of the blood, other simulations are based on the Windkessel effect, like Tsanas et al. (2009); Westerhof et al. (2009). The proposed simulator however relies on the standard Hagen-Poiseuille equation (Eq. 1) (Ganong et al., 2010; Marieb and Hoehn, 2010; Guyton and Hall, 2011) to calculate the volumetric flow rate ( $\phi_i$ ) in each blood vessel section  $i$  during a time step. The pulsatile work of the heart will impact the Eq. 1 by varying the pressure in a blood vessel section for a particular time step. The volumetric flow rate using the Hagen-Poiseuille equation is

$$\phi_i = \frac{\pi r_i^4 (P_i - P_{i+1})}{8\eta L_i} \quad (1)$$

where  $r_i$  and  $L_i$  are respectively the radius and the length of the  $i^{th}$  blood vessel section in which the blood flows.  $P_i$  and  $P_{i+1}$  are the pressure of the blood in these sections and  $\eta$  is the dynamic viscosity of the blood. The length of a blood vessel section remains constant through the simulation. The pulsatile flow produced by the heart must be damped. In the human body, the elasticity of the blood vessel is responsible for this damping. To mimic this behavior in the simulator, we propose a model inspired by Hook's law of elasticity. The difference in volume between the actual volume of the blood part and the relaxed volume of the blood vessel section replaces the displacement value in Hook's law. An adjusted elasticity constant ( $k$ ) is used, which can be specified for each blood vessel. The pressure  $P_i$  in the blood vessel section  $i$  is given by

$$P_i = \frac{k(V_i - W_i)}{2\pi r_i L_i} \quad (2)$$

where  $V_i$  is the volume of blood in the  $i^{th}$  vessel section and  $W_i$  is the relaxed (initial) volume of that blood vessel section. The resulting change in pressure at each time step influences the volumetric flow rate given by the Eq. 1 of the next time step.

During a time step ( $\Delta t$ ), all blood parts circulate throughout the sections of each blood vessel using the volumetric flow rate as explained previously. The new blood volume  $V'_i$  in each section  $i$  is represented with

$$V'_i = V_i - (\phi_i - \phi_{i-1})\Delta t \quad (3)$$

The blood part circulation is performed in reverse order. It is a design choice that required less memory than moving the blood parts in the way they flow. The simulator do not

have to keep an entire copy of each blood parts until the end of the update pass. The Fig. 3 shows the different steps to flow the blood through each sections of blood vessels. Each transferred blood part contains the same metabolites, in the same ratio, as the initial blood part they came from. Since volume changes at each time step, the blood flow is constantly recalculated.

As explained, the capillaries are modeled as a large container. In human body however, capillaries are large network of very small blood vessels. This arrangement of blood vessels induces a great resistance to blood flow due to the small radius of these vessels. The Eq. 1 can be rewrited as

$$\phi_i = \frac{(P_i - P_{i+1})}{R_i} \quad \text{with} \quad R_i = \frac{8\eta L_i}{\pi r_i^4} \quad (4)$$

where  $R$  is the resistance to blood flow of the blood vessel. In the simulator, since the capillaries are modeled as a large container, the resistance of the container must be adapted to represent more accurately the resistance of a network of blood vessels. Each capillaries container has a number of sub-vessels ( $n$ ). Each of them are identically modeled with a radius of 10 micrometers and a length proportional to the volume of blood of the entire capillaries container and the number of sub-vessel it contains. The model considers the sub-vessels in capillaries to be parallel, thus lowering the total resistance  $R_i$ , calculated with

$$\frac{1}{R_i} = \sum_{j=1}^n \frac{1}{R_j} \quad (5)$$

where  $R_j$  is the resistance in a sub vessel of the capillaries. Since all the  $R_j$  are identical, Eq. 5 can be simplified by

$$R_i = \frac{R_j}{n} \quad (6)$$

This model simplifies the blood flow in large and complex network of blood vessels in capillaries while keeping the physical incidence of their small radius on resistance.

To instill a pressure gradient to the bloodstream, the blood must be pumped. This role is devoted to the heart which is made of four parts. There are two atriums in which the blood arrives from the different circulation loops and there are two ventricles that pumped the blood out of the heart. The left atrium receives blood from the pulmonary loop while the right atrium receives it from the systemic loop. The simulated heart has also two group of self-polarizing cells, called sinoatrial node (SA node) and atrioventricular node (AV node). These nodes polarize and depolarize themselves to conduct the contraction of atriums and ventricles. For more details on heart nodes and their mechanisms, see (Guyton and Hall, 2011; Marieb and Hoehn, 2010). In the simulator, the polarization process goes through three different phases, as in reality. The first phase of the SA node is

the pacemaker. The pacemaker is a slow increase of the polarization of the node. The pacemaker phase is followed by a rapid depolarization until the maximum is reached. This abrupt depolarization emulates the sudden increase of ions (charged metabolites) that transfer through the membrane of the cells in a real heart. The contraction of the atriums happens at the end of that phase. Finally, the third phase is the repolarization until the minimum value is reached and the cycle restart. During the pacemaker phase, the atriums relax and retrieve their original volumes. The AV node follows the same process. However, when the SA node reaches its maximum polarization value, the node sends a signal to the AV node. That signal disturbs the pacemaker phase of the AV node and initializes the rapid polarization. The contraction of the ventricles occurs when the polarization of the AV node reaches its maximum value. The relaxation of the ventricle follows during the pacemaker phase of the AV node. This depolarization/polarization, which is only an exchange of charged metabolites (mainly of sodium and potassium) between the membrane of the cells forming the heart, is simplified for the simulator.

This level of details for the heart's implementation, using polarization levels, allows a better control over its reaction to external stimuli. Instead of using a predefined timer to conduct the heart's beat and trying to find the right value for it in the simulator, the hormonal and neuronal systems can increase or decrease the different value of polarization in the nodes to change the behavior of the heart allowing it to beat faster or slower.

## Respiratory system

The second simulated system is the respiratory system. This system is used to exchange the oxygen and the carbon dioxide between the body and the environment.

Like the human respiratory system, the virtual respiratory system has two main components, the lungs and the exterior environment. There are two lungs, and each of them is divided in different lobes. Each lobe contains alveoli in which the gases are exchanged with the blood in the capillaries. The lobes can be individually deactivated to simulate ill patients. The air enter the alveoli from the airways. The respiration control center is modeled as a timer. The rate of the respiration can be modified by changing the timer interval. It is in future plan to link this control center to a brain that will react to external stimuli, such as oxygen and carbon dioxide concentration as in real life. Fig. 4 shows the schematic of a lung in the system.

The air is modeled as an ideal gas (Eq. 7). And as a gas, it always fills all the available volume. The pressure  $P_{air}$  of the air depends on the volume  $V$  it fills, the temperature  $T$  in the lung, the quantity of gas ( $n$ ) and the ideal gas constant  $R$  with the relation

$$P_{air} = \frac{nRT}{V} \quad (7)$$

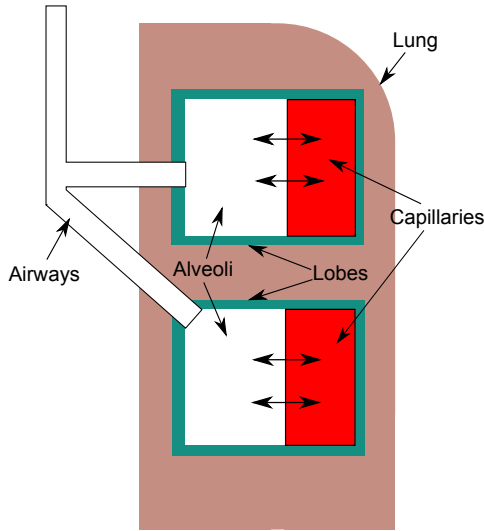


Figure 4: Schematic view of a simulated lung in the respiratory system. A lung is composed of lobes, each of them contains alveoli. The exterior environment is connected to the different alveoli using airways with one section. Capillaries of the circulatory system are linked to alveoli and exchange metabolites (i.e. oxygen and carbon dioxide) through gas diffusion.

The inspiration process increases the volume in the lung, while the expiration process decreases it. This difference of volume impacts the pressure of the air in the lungs, as explained by the Eq. 7. The air, as for the blood in the circulatory system, flows against its pressure gradient. But unlike the blood pressure, which depends of the elasticity of the blood vessels and the volume of blood it contains, the pressure of the air is calculated using the ideal gas equation (Eq. 7). In the simulator, the pressure of the exterior environment does not change as the respiration occurs. The air in the lung must always retrieve an equivalent pressure. Using Eq. 7, it is easy to find the amount of gas needed to balance the pressure between the lungs and the exterior environment. This amount of gas flows against the pressure gradient and balances the pressure in the lung at each time step.

The diffusion of gases between the blood and the alveoli of the lungs is driven by the partial pressure of these gases. However, each gas does not diffuse at the same rate. In the alveoli, all the gases composing the air are mixed together in a more complex gases mixture. This pressure of this mixture can be found using the Eq. 7. The partial pressure of each gas in the air can be calculated with the Dalton's law which states that the total pressure exerted by the mixture of non-reactive gases is equal to the sum of the partial pressures of each gases. For the air, the equation

$$p_i = P_{air} \frac{n_i}{n} \quad (8)$$

represents the partial pressure  $p_i$  of the  $i^{th}$  gas composing the air where  $n_i$  is the quantity in mole of this gas and  $n$  is the total amount of gases in the air. On the other hand, each gas dissolved in the blood has also a partial pressure. This partial pressure is calculated with the Henry's law stated as

$$q_i = \frac{n_i}{V} k_H \quad (9)$$

where  $q_i$  is the pressure of the  $i^{th}$  gas in the blood and  $n_i$  is the quantity of that gas in the blood.  $V$  is the volume of the blood part in which the gas is dissolved and  $k_H$  is the Henry's constant associated with the type of gas and the type of solution in which the gas is dissolved.

The diffusion of the gases takes place until the partial pressures in the air and in the blood are equal. Based on Fick's law of diffusion, the diffusion rate  $D_i$  of a gas  $i$  between the lung and the capillaries is

$$D_i = \frac{A(p_i - q_i)}{d} C_i \quad (10)$$

In this equation,  $C_i$  is the diffusion coefficient of the gas in the blood,  $A$  is the area of the blood vessel section that diffuses the gas,  $p_i$  and  $q_i$  are the partial pressure of that gas in the alveoli and in the capillaries and  $d$  is the distance of diffusion (Guyton and Hall, 2011). The quantity of gas  $Q_i$  added into the blood for a particular time  $\Delta t$  is

$$Q_i = D_i \Delta t \quad (11)$$

This diffusion process changes the respective partial pressure of oxygen in the blood and in the air of the alveoli. At the next time step, the diffusion rate changes accordingly and the cycle restarts upon equilibrium. For more information on gases diffusion in the human body, see Lumb (2010).

The respiratory system is responsible for the supply of new air into the body and for the expulsion of the exhausted one. In contrast with the circulatory system which is normally closed, the respiratory system is open. This particularity allows this system to be connected with different apparatus that provide breathable air or not. They are called the exterior environment. Normally, the respiratory system is connected to the atmosphere, composed at 78% of nitrogen and 21% of oxygen with the remaining being composed of many other compounds, like carbon dioxide and water vapor. The composition of this atmosphere influences the exchange of different gases in the lungs and in the body through the partial pressure of the composing gases. A higher concentration of oxygen in the air will increase the diffusion of this gas to the blood.

When the blood flows through the organs, it exchanges the oxygen and the carbon dioxide with them in a similar way than in the lungs. These exchanges change the partial pressure of these gases in the blood, resulting in continuous exchange when it passes through the lungs. The exchange



of gases in the organs follows the same principles as in the lungs with the equilibrium of partial pressures. The major difference is that the partial pressures of the gases in organs are found using the Henry's law (Eq. 9) instead of Dalton's law (Eq. 8). It is because the gases in organs are dissolved into the cells' fluids.

### Metabolites

All biological elements are called metabolites. Thus, every molecule that use the bloodstream or the pulmonary airways to circulate is considered to be a metabolite. It represents the oxygen, the carbon dioxide as well as the sodium, the potassium, the enzymes, the hormones and any other elements used by a system of the body. Like in human body, every blood parts, organs' fluid and air parts can contain metabolites. Instead of representing all the individual instance of a metabolite, like all atoms of oxygen dissolved in the blood, each metabolite is represented with a quantity representing the amount of individual instances. This simple representation of each set of metabolites in blood parts simplify the calculation in the different systems of the body. The pressure, the volume and the concentration, for example, can easily be found for a particular metabolite in a single blood part. The advantage of grouping all instances of the same metabolite limits the memory and the time needed to update all the systems. Furthermore, this simplification has only a small impact on the system, since it represents only a part of all instances of that metabolite in the whole body. The subdivision of the blood part and the air part allows precise control and limits actions to a specific section.

### Results

The developed system must be realistic enough to be used as a simulator for nurses education. The global behavior of the system must represents the way the human body behave in similar circumstances. The first experiment validates the behavior of the heart and the change in pressure into the bloodstream as the heart beats. It shows that cutting blood vessel to represent injuries has an impact on pressure. The second experiment validates the effect of the gases composing the atmosphere on the bloodstream. The experiment also demonstrates the effect of a ill lung to the respiratory system. The system needs 3 or 4 heart beats to stabilize at the beginning of a simulation.

### The Work of the Heart

The heart acts like a pump. It contracts and relaxes periodically. The effect of that pump is a continuous increase and decrease of the blood pressure in the arteries. The standard values of pressure for a healthy person are between 120 mmHg (or 16 000 Pascals) at the maximum and 60 mmHg (or 10 700 Pascals) at the minimum (Chobanian, 2004). These values represent the pressure in the blood vessels. It is the force exerted by the blood on a blood vessel wall. The

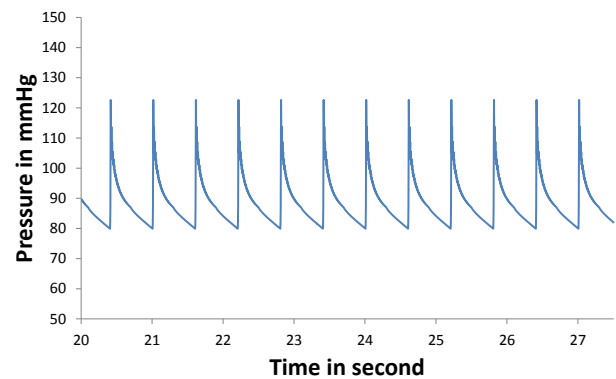


Figure 5: Pressure in the simulated aortic artery, at the exit of the heart. The pressure oscillates between 122 mmHg and 80 mmHg, which is in standard range for a main artery.

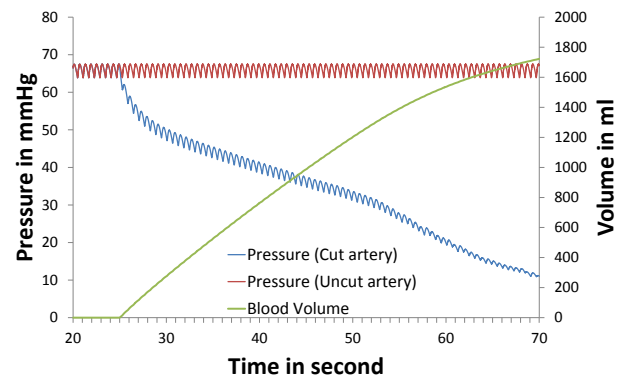


Figure 6: Blood pressure in a simulated artery that follows a cut. The green curve shows the volume of blood that has escaped through the cut. Red curve is the pressure in the artery when no cut is present in the system. Blue curve is the pressure in the same artery when a cut is present.

Fig. 5 shows the pressure of the blood in the simulated aortic artery at the exit of the left ventricle. The pressure rises when the heart contracts and decreases when the blood flows out of the heart.

To simulate an injured patient, the cardiovascular system allows blood vessels to be cut. The Fig 6 shows the effect of a cut at the end of the simulated artery network, before entering smaller capillaries vessels. Standard pressure in these arteries is lower since resistance and elasticity damped the pulse (Marieb and Hoehn, 2010). The volume of blood that leave the blood vessel is shown as well as the corresponding blood pressure in the next connected blood vessel. This cut to the artery should be deadly if no action is taken rapidly to mitigate the problem. The virtual patient rapidly loses blood, leading to a decrease in its pressure, and possibly death. The cut is considered open and the blood escap-

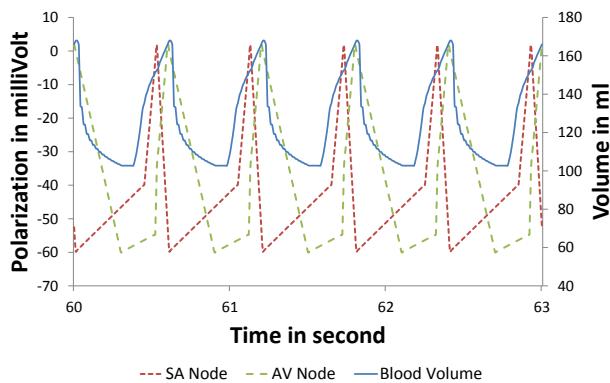


Figure 7: Blood volume variations of the left simulated ventricle over time. The first increase in blood represents the blood flowing from the left atrium to the left ventricle following the pressure gradient. The second increase occurs when the left atrium contracts, pushing the blood into the ventricle. The following decrease occurs when the ventricle contracts and expulses the blood into the aorta.

ing from the system exerts no pressure on the blood vessels or other organs. If the cut was modeled as a hemorrhage, the blood escaping the system would exert a pressure on the blood vessels, slowing the blood loss.

Another interesting feature of the simulator is the possibility to reproduce hypertension behavior. Results show that increasing the elasticity constant of a blood vessel, thus stiffening it, increases the maximum blood pressure in the neighboring vessels. Furthermore, the peak of the blood pressure in blood vessels occurs later in time with less elastic vessels, as explained in Mitchell (2006).

As explained previously, the heart is modeled as a pump with polarizations threshold. The pumping effect of the atriums and ventricles influences the volume of blood in the heart. The Fig. 7 shows the variations of the blood volume in the left ventricle of the simulated heart in relation with the polarization phases of the SA and AV nodes. The atriums contract when SA node reaches its maximum polarization value. The contraction of ventricles occurs shortly after when AV node reaches its own maximum polarization. The variation in blood volume for the ventricles is similar to the reality (Marieb and Hoehn, 2010).

### The Influence of the Air

Air composition influences the gas exchange in the human lungs. At the top of a mountain, the air pressure is lower than at sea level, which influences the partial pressure of oxygen. In the simulator, there is also a difference of partial pressure for the oxygen in the blood when the exterior atmospheric pressure changes. At sea level with standard atmospheric pressure (101 325 Pascals), the partial pressure of oxygen in the simulated blood is about 160 mmHg (104 mmHg in re-

ality) after passing the alveoli of the lungs. At 2000 meters of altitude (70 000 Pascals), the partial pressure of oxygen in the blood is about 110 mmHg (71.93 mmHg in reality) after passing the alveoli of the lungs. The difference of partial pressure of oxygen between the simulated blood and the reality is mainly due to the absence of water vapor in the air of the simulated lungs which increases the partial pressure of oxygen, following Eq. 8 (Marieb and Hoehn, 2010). The absence of water vapor simplifies the process of air exchange between the exterior environment and the simulated lungs. However, the results show clearly that a difference in initial atmospheric pressure impacts the system.

### Discussion

The goal of this work is to reproduce the behavior of a cardiopulmonary system. This simulator is used within a serious game for the training of nurses. One of the primary requirements of the system is a good representation of external and internal physiological processes. The system does not represent the exact reality. However, it must be realistic enough at a high level to create an immersive environment for the nurse. The presented simulator is based on simple mathematical concepts of chemistry and physics to mimic the basic interactions and behaviors of this complex system.

The presented approach, using a bottom-up design, relies on the principle of emergence to reproduce the complex interactions needed for this kind of simulation. This is in contrast with more standard approaches used in the video games industry. In a game, simulation and artificial intelligence often use finite-state machines. They are easy to define, the interaction between each component is clear and it can normally represent most of the desired behavior. However, relying on this model for a human body simulation has many disadvantages. First, there are many systems interacting together, thus complexifying the machine and increasing the chance to forget transitions when designing it. Second, this finite-state machine would required a huge amount of memory space and adding another system in the body simulation would require a lot of efforts to connect it with the others. Finally, every interactions on the model must be planned in the design stage, which are every actions and mistakes made by a nurse in training using the simulator. Naturally, predicting every mistakes and the order in which they will be made is a virtually impossible task. All these concerns have led us to create a simulator based on simple components interacting together so that complex behaviors can emerge.

The presented model is based on the subdivision of the entire system in logical units. The approach can be related to multi-agents system, where each part of the system executes its own job and send messages to other units to influence them. The model used for the simulator sends messages mainly through the bloodstream. An interesting effect of this message sending is the delay that occurs between the time a message is sent (i.e. an hormone is produced and re-

leased in the blood) and the time it reaches its zone of effect (i.e. when it binds to receptors to activate functions).

Again, the main advantage of using this kind of approach is the ability of the system to react automatically and adequately to the numerous possible actions of the nurses. When the simulator is in an unstable state, i.e. the virtual patient has injuries, the nurse in training must execute actions to stabilize it. When the nurse makes an action, the simulator must react adequately and it must continue running. The result of the action will impact the patient, thus reflecting what would happen in reality.

Mathematical models act in the same way as our approach. There is no need to plan every mistakes made by a nurse in the system. Actions will impact the formulas to provide new outputs. However, these models, even if they are extremely efficient and complete, can often be difficult to be divided and modeled as independent sub-system.

Modeling complex systems with simple components similar to multi-agents systems is an interesting idea that allows modularity, simplicity and speed of execution. The simpler formulas used in the presented simulator represent only basic physical interactions and are used to construct more complex behaviors. In a simulator that must reproduce global behavior instead of particular physiological principles, this modularity and simplicity of configuration offer a great advantage for both the developer of medical scenarios and the nurse in training.

## Conclusion

We describe a model of a cardiopulmonary system that is inspired by biological principles. The resulting behavior of the simulator corresponds to the actual behavior of a human body, thus allowing the simulator to be used for nurses' training. The decomposition into small and simple components to see the emergence of complex behaviors is an interesting way to model the problem. The teacher can specify injuries and illness to a patient, thus simplifying the creation of new medical scenarios.

## Acknowledgments

The research was supported in part by grants from the Mitacs Acceleration program. We also wish to thank the contribution of GolemLabs in the project, a video game company. Special thanks to all the GRITS (Research group in techno-pedagogy innovation in healthcare from University of Sherbrooke) members for their contribution to this work.

## References

- Attinger, E. O. and Anné, A. (1966). Simulation of the cardiovascular system. *Annals of the New York Academy of Sciences*, 128(3):810–829.
- Chobanian, A. V. (2004). The Seventh Report of the Joint National Committee on Prevention, Detection, Evaluation, and Treatment of High Blood Pressure. Technical report, US Department of health and human services.
- Clapworthy, G., Viceconti, M., Coveney, P. V., and Kohl, P. (2008). The virtual physiological human: building a framework for computational biomedicine I. Editorial. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 366(1878):2975–8.
- Coveney, P. V., Diaz, V., Hunter, P., Kohl, P., and Viceconti, M. (2011). The Virtual Physiological Human. *Interface Focus*, 1(3):281–285.
- Ellaway, R., Poulton, T., Fors, U., McGee, J. B., and Albright, S. (2008). Building a virtual patient commons. *Medical teacher*, 30(2):170–4.
- Ganong, W. F., Barrett, K. E., Barman, S. M., and Boitano, S. (2010). *Ganong's review of medical physiology*. McGraw-Hill, 23rd edition.
- Guyton, A. C. and Hall, J. E. (2011). *Textbook of medical physiology*. Saunders/Elsevier, 12th edition.
- Hansen, M. M. (2008). Versatile, immersive, creative and dynamic virtual 3-D healthcare learning environments: a review of the literature. *Journal of medical Internet research*, 10(3):e26.
- Hester, R. L., Brown, A. J., Husband, L., Iliescu, R., Pruett, D., Summers, R., and Coleman, T. G. (2011). HumMod: A Modeling Environment for the Simulation of Integrative Human Physiology. *Frontiers in physiology*, 2(April):12.
- Issenberg, S. B., S. Gordon, M., Gordon, D. L., E. Safford, R., and R. Hart, I. (2001). Simulation and new learning technologies. *Medical teacher*, 23(1).
- Lumb, A. B. (2010). *Nunn's Applied Respiratory Physiology*. Elsevier, 7th edition.
- MacIntyre, N. R. (2004). Respiratory system simulations and modeling. *Respiratory care*, 49(4):401–8; discussion 408–9.
- Marieb, E. N. and Hoehn, K. (2010). *Human anatomy & physiology*. Benjamin Cummings, 8th edition.
- Mitchell, G. F. (2006). Triangulating the peaks of arterial pressure. *Hypertension*, 48(4):543–5.
- Morgan, P. J., Cleave-Hogg, D., Desousa, S., and Lam-McCulloch, J. (2006). Applying theory to practice in undergraduate education using high fidelity simulation. *Medical teacher*, 28(1):e10–5.
- Tsanas, A., Goulernas, J. Y., Vartela, V., Tsiapras, D., Theodorakis, G., Fisher, A. C., and Sfirakis, P. (2009). The Windkessel model revisited: a qualitative analysis of the circulatory system. *Medical engineering & physics*, 31(5):581–8.
- Westerhof, N., Lankhaar, J.-W., and Westerhof, B. E. (2009). The arterial Windkessel. *Medical & biological engineering & computing*, 47(2):131–41.
- Zary, N., Johnson, G., Boberg, J., and Fors, U. G. H. (2006). Development, implementation and pilot evaluation of a Web-based Virtual Patient Case Simulation environment–Web-SP. *BMC medical education*, 6:10.

## Artificial Causal Space-Time

Yukio-Pegio Gunji<sup>1,2,\*</sup>, Tomoko Sakiyama<sup>1</sup>, Sohei Wakisaka<sup>3</sup>, Naotaka Fujii<sup>3</sup> and Tomoaki Nakamura<sup>1</sup>

<sup>1</sup>Department of Earth & Planetary Science, Faculty of Science, Kobe University, Nada Kobe 657-8501 Japan

<sup>2</sup>The Unconventional Computing Centre, University of the West England, Bristol, BS16 1QY, UK

<sup>3</sup>Brain Research Institute, RIKEN, Hirosawa 2-1, Wako, Saitama, 351-0198, Japan

\*yukio@kobe-u.ac.jp

### Abstract

If a space-time with a causal relationship is viewed from an observer in space-time, the interaction between space-time and the observer has to be implemented. We describe this interaction using a pair of causal sets and its semantics or using a pair of Point and Open Logic. We here propose an artificial causal space-time called an evolutionary topological system, which is based on the changeability between logical operations (disjunction and conjunction) and logical elements (join and meet) of a causal set. The conflict resulting from the interaction is locally and temporally removed by replacing disjunction and join (or conjunction and meet), and a causal set is verified to evolve to a particular logical structure based on the simple summation. We also show this model can design an abnormal space-time feeling, such as an out-of-body experience.

### Introduction

One of the most intriguing and important models for subjective and/or cognitive time was proposed by a philosopher, McTaggart (1908). He evaluated two model types, called the A series and B series. The B series consists of events that are linearly ordered and designated by “before” and “after”. The A series consists of past, present and future events that cannot co-exist and are exclusive of each other. Although McTaggart himself argued that neither the A nor B series can be a model for time, the A and/or B series are still used as models for time in philosophy (Grey, 1997; Mellor, 1998; Gunji et al., 2009).

Although the original A and B series appear to be too speculative to be considered models for time, a pair of the A and B series can be utilized as a causal set (Bombelli et al., 1987) and its semantics in the field of quantum mechanics (Markopoulou, 2000), independent of philosophy. The B series corresponds to a causal set defined by a partially ordered set. The A series corresponds to a sieve in the semantics of a causal set. Thus, the idea of the A and B series is taken as a causal relationship and can be argued in quantum gravity (Klugry and Sepanina, 2011).

A causal set that serves as a model for causal relationships in space-time is a given for an observer. It is assumed that an observer living in a space-time passively observes, memorizes and recalls a series of events and calls that set of events the past, present or future, depending on the location of the observer. Although Markopoulou introduced the stance of an observer who observes a space-time internally, the

interaction between an observer and space-time was not described. The role of an observer still remains.

We here propose an artificial causal space-time in which an observer moving in a space-time can interact with space-time itself. Why does the interaction occur? Independent of the idea of a causal set, Vickers (1996) proposed the generalized idea of a causal set and its semantics in the form of a topological system. In this framework, logical operations are defined in a causal set, called Point Logic, and in its semantics, called Open Logic. Because a generalized causal set and its semantics are related to each other by a particular binary relationship, they are restricted with respect to each other. It entails a conflict between Point and Open Logic that has to be resolved, which is why the interaction between Point Logic (in a causal set) and Open Logic (in its semantics) can occur.

Against a conflict, Vickers introduced a limited logical operation that is found as ubiquitously as Scott topology (Scott, 1976). In other words, his solution to resolve a conflict results from the stance of an observer who sees a space-time externally. Once his solution is introduced, a conflict never occurs. Therefore, in principle, there is no interaction between an observer and space-time.

Our artificial causal space-time is a dynamic causal set equipped with a particular rule by which perpetually generated conflicts between a causal set and its semantics are discarded locally and temporally. Even if a local consistency is generated, removing the local conflict can generate another local conflict. This dynamic evolves a causal set toward a particular system in which the cause-effect relationship can be calculated by summing up the causes. We also show that the interaction of Point and Open logic can generate a particular subjective sensation called an “out-of-body experience (OBE)” that seems to differ from the OBE previously reported (Ehrsson, 2007; Lenggenhager et al., 2007). Our model suggests how to create new sensations and emotion in subjective space-time.

### Causal sets and Topological System

Imagine a series of events in time, such as  $\dots \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow \dots$ , and you are at event  $x_2$ . The future and past of  $x_2$  can be expressed as the set of  $\{\dots, x_1, x_2\}$  and  $\{x_2, x_3, \dots\}$ , respectively, and the present at event  $x_2$  is expressed as the intersection of the future and past as  $\{x_2\}$ . This idea is the



essence of the B and A series presented by McTaggart (Gunji et al., 2010).

A causal set is a set of events defined as a partially ordered set. Its semantics is a collection of, for example, all possible futures. This pair can be a generalized pair of the B and A series. In addition, the introduction of the relationship between a causal set and its semantics derives motivation for the interaction between a causal set and its semantics.

### Causal set and its semantics

**Causal set and space-time.** A causal set consists of separable events. Each event can be connected to another event via a directed edge without loops. If two events are connected by two edges that have different directions, they are equivalent to each other. Thus, this particular directed network can be expressed as a partially ordered set (POS) (Davey and Priestley, 2002). If an event and a directed edge are expressed as an alphabet and  $\leq$ , respectively, the POS satisfies the following: (i)  $a \leq a$ , (ii)  $a \leq b$  and  $b \leq a$  implies  $a = b$ , and (iii)  $a \leq b$  and  $b \leq c$  implies  $a \leq c$ .

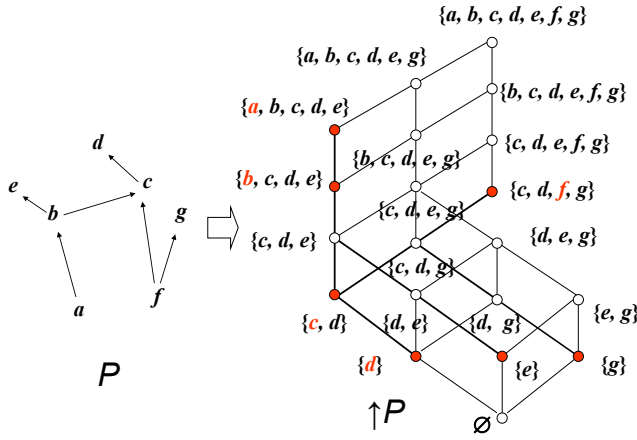


Figure 1. A causal set,  $P$ , and its semantics,  $\uparrow P$ . In a causal set, partial order,  $\leq$ , is expressed as an arrow. In  $\uparrow P$  an element of  $\uparrow P$  (i.e., a subset of  $P$ ) is expressed as a circle. If  $A \subseteq B$ , they are connected by a line with upper  $B$  and lower  $A$ . A filled circle represents  $\uparrow p$  with  $p$  in  $P$ .

Some terminologies are also added. Any elements  $a$  and  $b$  in a POS,  $P$ , are anti-chain with each other if neither  $a \leq b$  nor  $b \leq a$  holds. For any subsets,  $Q \subseteq P$  join of  $Q$ , denoted by  $\vee Q$ , is defined such that for any  $q \in Q$ ,  $q \leq \vee Q$  and if  $q \leq s$ , then  $\vee Q \leq s$ . In particular, if  $Q$  is a two-element set such as  $\{a, b\}$ ,  $\vee \{a, b\}$  is represented by  $a \vee b$ . Similarly, the meet of  $Q$ , denoted by  $\wedge Q$ , is defined such that for any  $q \in Q$ ,  $q \geq \wedge Q$  and if  $q \geq s$ , then  $\wedge Q \geq s$ . If  $Q$  is a two-element set such as  $\{a, b\}$ ,  $\wedge \{a, b\}$  is represented by  $a \wedge b$ . Given a partially ordered set,  $P$ , if for any  $x, y \in P$ ,  $x \wedge y, x \vee y \in P$ , then  $P$  is called a lattice.

Given a POS,  $P$ , for any  $p \in P$ , the future of  $P$  is defined by  $\uparrow p = \{x \in P \mid p \leq x\}$ . The semantics of  $P$  is a collection of the possible unions of all  $\uparrow p$  for any element  $p$  in  $P$ . Thus, it is defined by  $\uparrow P = \{\uparrow Q \mid Q \subseteq P\}$  where  $\uparrow Q = \{y \in P \mid (\exists x \in Q) y \geq x\}$ . Fig. 1 shows an example of a causal set,  $P$ , and  $\uparrow P$ .

In  $P$ , the future of  $b$  is expressed as  $\uparrow b = \{b, c, d, e\}$ . Similarly,  $\uparrow c = \{c, d\}$ , and then  $\uparrow c \subseteq \uparrow b$ . Any elements other than  $\uparrow p$  (with

$p$  in  $P$ ) in  $\uparrow P$  are expressed as a union of the  $\uparrow p$ 's (with  $p$  in  $P$ ), such as  $\{b, c, d, e, f, g\} = \uparrow b \cup \uparrow f$ . Tracing the filled circle in  $\uparrow P$ , one can see that the ordered structure of  $P$  is embedded in  $\uparrow P$ .

In the context of a causal set, there is no discussion about the relationship between a causal set and its semantics. The relationship is introduced in the context of formal logic, independent of the idea of a causal set. It is called a topological system.

**Topological system.** Given a set  $S$ , if a collection of subsets of  $S$  satisfies an axiom of opens, it is called a topology or topological space. An axiom of opens is the following: (i)  $S$  and empty set are opens, (ii) a finite intersection of opens is an open, and (iii) a union of opens is an open. A power set that is a collection of all subsets of  $S$  is the densest topology, and a collection consisting only of  $S$  and empty is the sparsest topology. Topology is a type of metric that can be used to recognize a set,  $S$ .

Because topology is constrained under a particular axiom of opens, Vickers (1996) attempted to generalize a topology in the form of a binary relationship between a collection of points and a collection of opens, which he called a topological system. A collection of points and opens is defined by a triplet,  $\langle P, L, R \rangle$ , where  $P$  is a set,  $L$  is a locale and  $R$  is a binary relationship between  $P$  and  $L$ . A locale is a partially ordered set that is closed with respect to union (disjunction),  $\cup$ , and finite intersection (conjunction),  $\cap$ , and that satisfies the distributive law, such that for any  $a, b$ , and  $c \in L$ ,  $a \cap (b \cup c) = (a \cap b) \cup (a \cap c)$ . It is trivially true that a triplet  $\langle P, \uparrow P, \in \rangle$  also satisfies the definition of a topological system.

Because a locale contains logical operations, we can define logical operations for opens. What about points? Imagine an observer moving in a causal set,  $P$ . He can manipulate logical operations for an element of  $\uparrow P$ , which is derived from a basic assumption in which an observer discriminates an event from other events and recognizes a set created from  $P$ . Thus, he can also manipulate a logical operation for events. It is, therefore, reasonable that a logical operation for points is also introduced.

### Point Logic and Open Logic

**Definition of Point and Open Logic.** Given a causal set, its semantics and the binary relationship between them, such as the triplet  $\langle P, L, R \rangle$ , Vickers (1996) introduced logical operations in both  $P$  and  $L$ . For  $S$ , a subset of a local, logical operation, conjunction (AND)  $\cap$  and disjunction (OR) are defined by

$$xR \cap S :\Leftrightarrow (\forall a \in S) xRa \quad (1)$$

$$xR \cup S :\Leftrightarrow (\exists a \in S) xRa \quad (2)$$

Because a locale is a generalization of an open set in topological space, logic in a locale is called Open Logic.

Similarly, for  $T$ , a subset of  $P$ , conjunction and disjunction are defined by

$$\cap Tra :\Leftrightarrow (\forall x \in T) xRa \quad (3)$$

$$\cup Tra :\Leftrightarrow (\exists x \in T) xRa \quad (4)$$

Because an element of a set is a point, logic in  $P$  is called Point Logic. These logical operations can also be defined in  $\langle P, \uparrow P, \in \rangle$  by using  $\in$  instead of  $R$ .

**Conflict between Point and Open Logic.** Although Point Logic is defined for a point and Open Logic is defined for a set, logical operations are defined in the same manner, and they are related by a binary relationship, which entails a conflict between Point and Open Logic. The question arises as to how we can resolve this conflict.

Let us consider a causal system,  $\langle P, \uparrow P, \in \rangle$ . Given a partially ordered set  $P$ , the followings two are trivially true:

$$a \cup b \in \{a\}, \quad a \cup b \in \{b\}. \quad (5)$$

Because the left-hand term means  $a$  or  $b$  even if the right-hand set contains either  $a$  or  $b$ , then the statement holds. Thus, statement (5) holds because of definition (4). Statement (5),  $a \cup b \in \{a\}$  and  $a \cup b \in \{b\}$ , means that  $a \cup b$  belongs to both  $\{a\}$  and  $\{b\}$ . In other words,  $(\forall a \in S) x \in a$  where  $x = a \cup b$  and  $S = \{\{a\}, \{b\}\}$ . Due to definition (1), we obtain

$$a \cup b \in \{a\} \cap \{b\}. \quad (6)$$

However, statement (6) also means that  $(a \in \{a\} \text{ and } a \in \{b\})$  or  $(b \in \{a\} \text{ and } b \in \{b\})$ ; this statement thus never holds. This type of statement results from the conflict between Point and Open Logic.

A solution proposed by Vickers is to restrict an operation of disjunction. In statement (6), disjunction is applied to a set  $\{a, b\}$ , which results in  $\cup\{a, b\} = a \cup b$  in the left-hand term. That result entails a conflict. In Vicker's solution, disjunction can be operated only to a directed set,  $D$ , which is defined by the following: for any  $x, y \in D$ , there exists  $z \in D$  such that  $x \leq z$  and  $y \leq z$ . Note that  $\{a, b\}$  is not a directed set because  $a$  and  $b$  are anti-chains for each other. Why are they anti-chains? If  $a \leq b$  or  $b \leq a$ , the right-hand set can be replaced by  $\{a, b\} \cap \{b\}$  in the case of  $a \leq b$ , or  $\{a\} \cap \{a, b\}$  in the case of  $b \leq a$ , because any sets in the right side of  $\in$  have to be an element of  $\uparrow P$ .

Because  $a \leq a \vee b$  and  $b \leq a \vee b$ ,  $\{a, b, a \vee b\}$  is a directed set. In considering the case in which a join  $a \vee b$  exists and  $\cup\{a, b, a \vee b\}$ , the right-hand term in (6) has to be replaced by  $\{a, a \vee b\} \cap \{b, a \vee b\}$ . Therefore, we obtain

$$a \cup b \cup a \vee b \in \{a, a \vee b\} \cap \{b, a \vee b\}. \quad (7)$$

Thus, at least one of  $a, b$  or  $a \vee b$  belongs to both  $\{a, a \vee b\}$  and  $\{b, a \vee b\}$ , which is why statement (7) holds. When disjunction is applied only to a directed set, this particular disjunction is called a directed disjunction and is represented by  $\cup^\uparrow$ . Thus, definition (4) in a causal system  $\langle P, \uparrow P, \in \rangle$  is replaced by

$$\cup^\uparrow T \in a : \Leftrightarrow (\exists x \in T) x \in a, \quad (8)$$

where  $T$  is a directed set. Inversely, conjunction in Point Logic related to disjunction in Open logic entails a conflict that can be resolved by directed disjunction in Open Logic.

This type of solution to resolve a conflict is the construction of a one-to-one correspondence between Point and Open Logic by discarding parts that cannot be correspondent with each other. Although one-to-one correspondence is achieved,

logical operations are restricted and used incompletely. Alternatively, we here intend to propose a solution to resolve the conflict in which logical operation is not restricted.

## Causal dynamics with local consistency

How can we resolve the conflict between Point and Open Logic? Instead of introducing directed disjunction, we here introduce the changeability of join and disjunction and of meet and conjunction. Actually, if we can replace disjunction with join in  $P$  whenever we use disjunction, the conflict between Point and Open logic can be resolved. For example, considering statement (6), one can obtain

$$a \vee b \in \{a, a \vee b\} \cap \{b, a \vee b\}. \quad (9)$$

Thus, by manipulating  $a$  and  $b$ , there exists a join of  $a$  and  $b$ , and  $a \cup b$  can be replaced by  $a \vee b$ . The conflict can be resolved by this changeability of disjunction and join and of conjunction and meet, which is why the changeability can be interpreted as a local consistency between Point and Open Logic.

The changeability of conjunction and meet can be verified, but that of disjunction and join cannot be proved. To create a causal system  $P$  that satisfies the changeability of disjunction and join,  $P$  is locally modified by a particular rule that is a local modification based on the dynamics of a causal system. First, we show the verification of the changeability.

### Changeability of logical operation and element in $P$

**Conjunction and meet.** Because each element of  $\uparrow P$  is expressed as an upper set of  $P$ , the changeability of conjunction and meet is expressed via  $\uparrow x$  with  $x \in P$ .

#### Proposition 1 (Changeability of conjunction and meet)

Given a topological system  $\langle P, \uparrow P, \in \rangle$ , for any  $a, b, x \in P$ ,  $a \cap b \in \uparrow x \Leftrightarrow a \wedge b \in \uparrow x$  if there exists  $a \wedge b$  for  $a$  and  $b$ .

**Proof.** (i) Assume  $a \cap b \in \uparrow x$ . It means that  $a \geq x$  and  $b \geq x$  and then  $x$  is a lower bound for  $\{a, b\}$ . Because of the meet, the greatest lower bound  $a \wedge b$  is larger than  $x$ ,  $a \wedge b \in \uparrow x$ . We verified  $a \cap b \in \uparrow x \Rightarrow a \wedge b \in \uparrow x$ .

(ii) Assume  $a \wedge b \in \uparrow x$ . We obtain  $a \wedge b \geq x$ . Because  $a \geq a \wedge b$  and  $b \geq a \wedge b$ , we obtain  $a \geq x$  and  $b \geq x$ , which means  $a \cap b \in \uparrow x$ .

#### Proposition 2 (Semi-changeability of disjunction and join)

Given a topological system  $\langle P, \uparrow P, \in \rangle$ , for any  $a, b, x \in P$ ,  $a \cup b \in \uparrow x \Rightarrow a \vee b \in \uparrow x$  if there exists  $a \vee b$  for  $a$  and  $b$ .

**Proof.** Assume  $a \cup b \in \uparrow x$ . It means that  $a \geq x$  or  $b \geq x$ . Because  $a \vee b \geq a$  and  $a \vee b \geq b$ ,  $a \vee b \geq x$  always holds. Thus, we obtain  $a \vee b \in \uparrow x$ .

The inverse of proposition 2, in which  $a \vee b \in \uparrow x \Rightarrow a \cup b \in \uparrow x$ , never holds. A counterexample is given by a partially ordered set,  $\{a, b, x, a \vee b\}$ , where  $a \vee b \geq x$  and  $a, b$  and  $x$  are anti-chain with each other. Although  $a \vee b \in \uparrow x$  holds in this partially ordered set,  $a \geq x$  or  $b \geq x$  never holds because they are anti-chains. Thus,  $a \vee b \in \uparrow x \Rightarrow a \cup b \in \uparrow x$  does not hold in

general. Therefore, the changeability of disjunction and join never holds in any partially ordered set.

Thus, we define a particular dynamic by which the changeability of disjoint and join is locally implemented.

## Dynamical system for local consistency

We implement a dynamic for the changeability of disjunction and join and simulate the time development of an evolutionary causal set. For this purpose, we define a causal set consisting of binary sequences.

**Definition 3 (Causal set of binary sequences)** A causal set,  $P$ , of binary sequences consists of  $n$  bits of sequence,  $\mathbf{a} = \langle a_1 a_2 \dots a_n \rangle$  where each  $a_k = 0$  or 1 for any  $k \in \{1, 2, \dots, n\}$ . The order is  $\mathbf{a} \leq \mathbf{b}$  if  $a_k \leq b_k$  for any  $k \in \{1, 2, \dots, n\}$ .

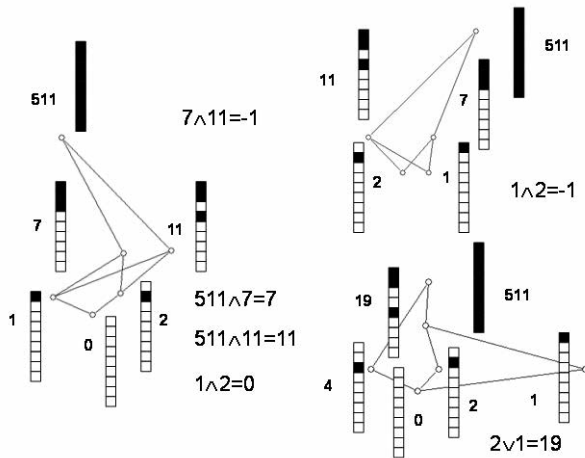


Figure 2. Some examples of a causal set of binary sequences. Each element represented by a binary column is expressed as a decimal number. If there is no join or meet, -1 is returned.

Fig. 2 shows three examples of a causal set of binary sequences. Only the right lower set is a lattice, and the others are not. A binary column represents a binary sequence, where the black and white squares represent 1 and 0 digits, respectively. A decimal number is representation of a binary sequence. Join and meet for a two-element set are shown as an example.

Notice that join is not a union of binary sequences and that meet is not intersection. Here, we denote a union by the symbol  $\oplus$ . For a pair of binary sequences,  $\mathbf{a}$  and  $\mathbf{b}$ ,  $\mathbf{a} \oplus \mathbf{b} = \langle \dots a_k + b_k \dots \rangle$  where  $0+0=0$ ,  $0+1=1+0=1+1=1$ . If a union  $\oplus$  is applied to  $\{1, 2\}$  in the causal set in the lower right of Fig. 2, we obtain  $1 \oplus 2 = 3$ . Because there is no binary sequence represented by 3 in this causal set, we obtain  $1 \vee 2 = 19$ . Similarly, meet is not an intersection. Here, we denote intersection by the symbol  $\otimes$ . For  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{a} \otimes \mathbf{b} = \langle \dots a_k b_k \dots \rangle$  where  $0 \times 0 = 0 \times 1 = 1 \times 0 = 0$ ,  $1 \times 1 = 1$ . If an intersection is applied to  $\{11, 7\}$  in the upper right of Fig. 2,  $11 \otimes 7 = 3$  is obtained. However, because there is no 3, we then obtain  $11 \wedge 7 = -1$  (i.e., there is no meet).

Given a causal set of binary sequences, we introduce the dynamics of a causal set and define an evolutionary topological system.

**Definition 4 (Evolutionary topological system)** An evolutionary topological system is defined by  $\langle P^i, \uparrow P^i, \in, F \rangle$ , where  $P^0$  is an initial causal set of  $n$  bit sequences that are randomly given. The time development of a causal system is defined by

$$P^{t+1}=F(P^t) \quad (10)$$

where  $F(P^t)$  is defined by the following:

- (i) Randomly choose two elements  $\mathbf{a}$  and  $\mathbf{b}$  from  $P'$  and obtain  $\uparrow \mathbf{a}$  and  $\uparrow \mathbf{b}$  from  $\uparrow P'$ .
- (ii) If the statement  $\mathbf{a} \wedge \mathbf{b} \in \uparrow \mathbf{a} \cup \uparrow \mathbf{b}$  does not hold, then calculate  $\mathbf{a} \wedge \mathbf{b}$ .
- (iii) If  $\mathbf{a} \wedge \mathbf{b}$  does not exist, calculate  $\mathbf{a} \otimes \mathbf{b}$ , and then each bit with a 1 value is replaced by 0 or nothing happened with an equal probability. When this replacement is denoted by  $\text{RAND}_1$ , we add the new element of

$$a \wedge b = \langle \dots \text{RAND}_1(a_k \times b_k) \dots \rangle \quad (11)$$

- (iv) If the statement  $a \cup b \in \uparrow a \cap \uparrow b$  does not hold, then calculate  $a \vee b$ .
- (v) If  $a \vee b$  does not exist, calculate  $a \oplus b$ , and then each bit with a 0 value is replaced by 1 or nothing happened with an equal probability. When this replacement is denoted by  $\text{RAND}_0$ , we add the new element of

$$a \vee b = \langle \dots \text{RAND}_0(a_k + b_k) \dots \rangle \quad (12)$$

- ```

to  $P^j$ .
(vi) Choose  $c$  from  $P^j$ 
(vii) If  $(a \vee b \geq c)$  {
        if  $(a \geq c$  or  $b \geq c)$  {
        } else {
            if  $(c \wedge a$  and  $c \wedge b$  are anti-chain) {
            } else {
                Remove  $c$  from  $P^j$ 
                Add  $x$  such that  $a \geq x$ 
            }
        }
    }
}

```

The dynamics of an evolutionary topological system are based on the changeability of conjunction and meet and of disjunction and join. According to Proposition 1, if the meet of  $\mathbf{a}$  and  $\mathbf{b}$  exists, then it can be replaced by the conjunction of  $\mathbf{a}$  and  $\mathbf{b}$ . Thus, the meet of  $\mathbf{a}$  and  $\mathbf{b}$  is generated for the changeability of meet and conjunction. Notice that, due to  $\text{RAND}_1$ , a generated meet is not an intersection.

In contrast, given Proposition 2,  $a \vee b \in \uparrow x \Rightarrow a \cup b \in \uparrow x$  does not hold even if the join of  $a$  and  $b$  exists. As mentioned previously, in a causal set  $\{a, b, c, a \vee b\}$  where  $a \vee b \geq c$  and  $a, b$  and  $c$  are anti-chain with each other,  $a \vee b$  cannot be replaced by  $a \cup b$ . Thus, this type of case is removed. This procedure is implemented by “Remove  $c$  from  $P$ ”. However, in the case of  $P' = \{a \leq c \leq a \vee b, b\}$ , if  $c$  is removed and  $x$  is added such that  $x \leq a$ .

a causal set with the same structure,  $P^{t+1} = \{x \leq a \leq x \vee b, b\}$ , is obtained. Thus, this process has fallen into infinite regression. Actually, in a causal set of binary sequences with a finite length, this process turns a causal set degenerate into a one-point set of the least element,  $0$ , which is why the procedure (vii) discards the case of the subset,  $\{a \leq c \leq a \vee b, b\}$ , from removing the  $c$  procedure

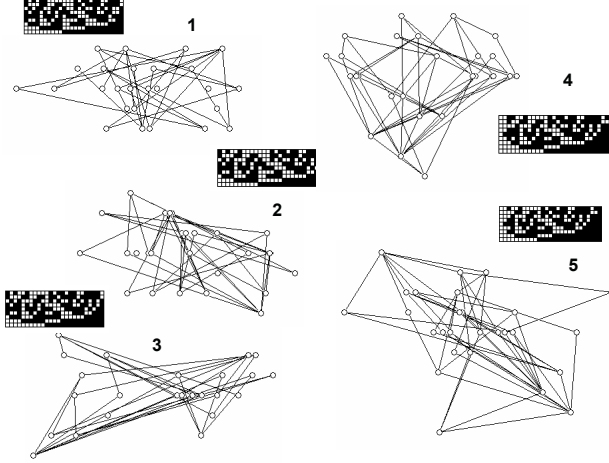


Figure 3. Time development of an evolutionary topological system.  $P^1, P^2, \dots, P^5$  of 9-bit sequences are shown in the form of a Hasse diagram. All elements of  $P^t$  are represented by binary columns above  $P^t$ .

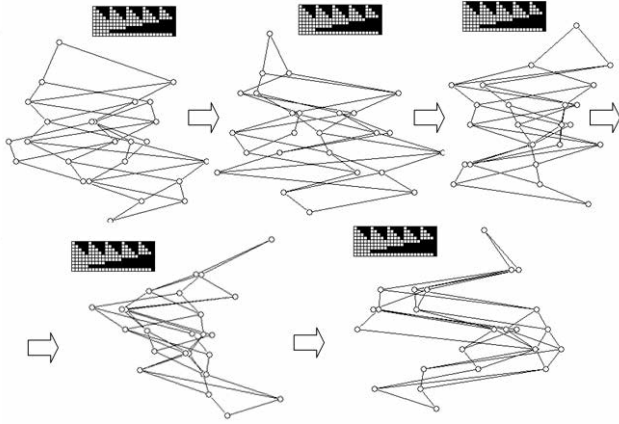


Figure 4. A causal set of an evolutionary topological system has evolved into a distributive lattice.

Fig. 3 shows the time development of an evolutionary topological system. Each causal set,  $P^t$ , is represented in the form of a Hasse diagram (i.e., if  $a \leq b$  and there is no other element between  $a$  and  $b$ ,  $a$  and  $b$  are connected by a line). The initial causal set is randomly given, and  $P^{t+1} = F(P^t)$  is iterated.

Various time developments suggest that the dynamics defined in definition 4 create a causal set with a particular structure, a distributive lattice. If there is no meet or join, they

are added to the causal set. Thus, a POS that is not a lattice is changed into a lattice. In particular, once a causal set becomes a distributive lattice, it is not changed again, and the structure is maintained (Fig. 4).

A distributive lattice is verified to be a lattice that contains no  $M_3$  and  $N_5$  as a sub-lattice (a subset of a lattice that is closed with respect to a join and meet). The structure of  $M_3$  and  $N_5$  are shown in Fig. 5. Given  $M_3$  and  $N_5$  as the initial causal sets, dynamics modify the causal set into a lattice without  $M_3$  and  $N_5$ . As a result, the causal set evolves into a distributive lattice.

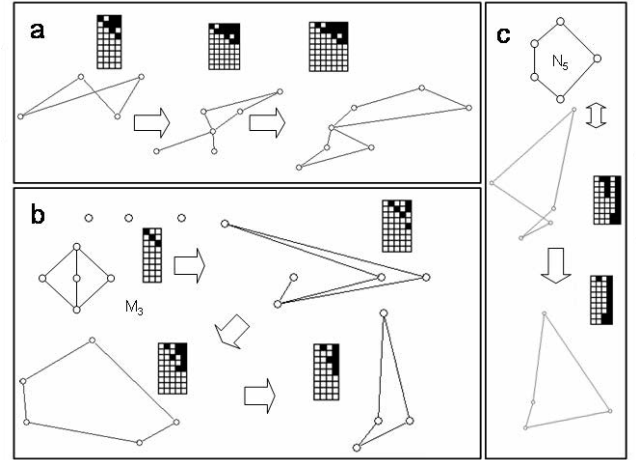


Figure 5. a. The initial POS is developed into a distributive lattice (DL). b.  $M_3$  has also fallen into DL. The arrow represents time development.

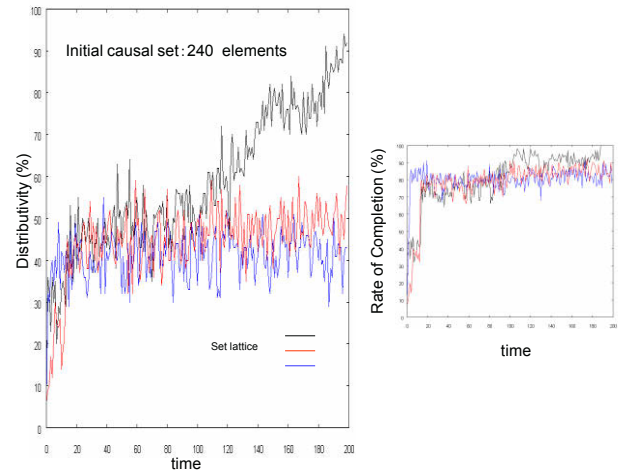


Figure 6. Distributivity plotted against time for a causal set (left) and the rate of completion against time (right). The black line corresponds to an evolutionary topological system, and the red and blue lines correspond to the control experiments. The initial set consists of 240 binary sequences.

To estimate how a causal set converges into a distributive lattice, we define distributivity for a causal set  $P^t$ : three elements  $a, b$  and  $c$ , are randomly chosen, and whether  $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$  is evaluated, as long as the required meet and join exist, for  $K$  times. Distributivity is defined by



the number of equality divided by  $K$ . We also define two control experiments to compare with the evolutionary topological system. The first control dynamic is only the application of completion (i.e., the procedure of adding meet and join) to a causal set,  $P'$ . It does not contain the procedure of (vii) in definition 4. The second control dynamic also does not contain (vii), and the completion is indeed defined by a union and intersection. Thus, the second control dynamic does not contain equations (11) and (12); therefore,  $a \wedge b = a \otimes b$ , and  $a \vee b = a \oplus b$ . If completion is achieved by these procedures, a causal set can become a lattice of sets, which is well known as a distributive lattice.

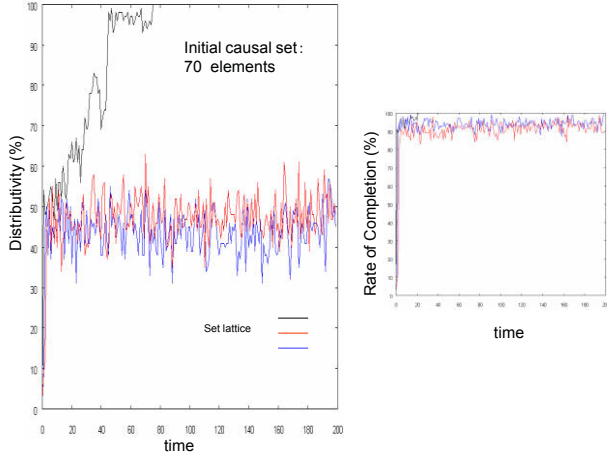


Figure 7. Distributivity plotted against time for a causal set (left) and the rate of completion against time (right). The black line corresponds to an evolutionary topological system, and the red and blue lines correspond to the control experiments. The initial set consists of 70 binary sequences.

As shown in Fig. 6, the distributivity of a causal set of an evolutionary topological system increases towards a distributive lattice. A causal set consists of 9-bit sequences and 240 elements (sequences) initially. Adding the meet and join and removing elements that do not satisfy the changeability increases the distributivity. Compared with the evolutionary topological system, the two control experiments never increase their distributivity. Even if a meet and join are added in the form of an intersection and union, respectively, in the second control dynamic, adding a new element as the join and/or meet entails another requirement to create the join and/or meet. Thus, completion cannot be achieved, and the distributivity is not increased. If the number of initial causal sets is small, the tendency of increasing distributivity is also found (Fig. 7).

### Distributivity of a causal set

An evolutionary topological system can converge into a distributive lattice. Because of the restricted changeability of join and disjunction and of meet and conjunction, when the system reaches a distributive lattice can be verified.

**Proposition 5 (Evolutionary topological system)** An evolutionary topological system defined by  $\langle P', \uparrow P', \in, F \rangle$  will converge into a distributive lattice.

**Proof. (i)** The case is that  $b \vee c \geq a$  and  $a \wedge b$  is an anti-chain of  $a \wedge c$  but not  $b \geq a$  or  $c \geq a, b$  (i.e.,  $b \vee c \in \uparrow a \Rightarrow b \cup c \in \uparrow a$  does not hold, but procedure (vii) in definition 4 is not applied). This case is shown in Fig. 8. This case leads to a distributive sub-lattice by adding the meet,  $b \wedge c$ , which can be achieved at any point.

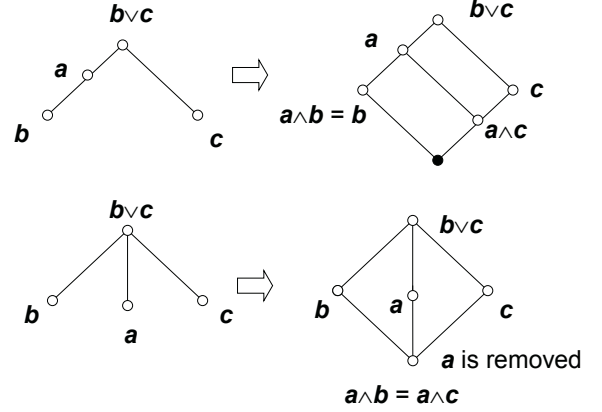


Figure 8. The case that  $b \vee c \geq a$  and that  $a \wedge b$  is an anti-chain of  $a \wedge c$  but not  $b \geq a$  or  $c \geq a, b$  (above). The case that  $b \vee c \geq a$  and that  $a \wedge b$  is not an anti-chain of  $a \wedge c$ ,  $b \geq a$  or  $c \geq a$  (below).

(ii) The case that  $b \vee c \geq a$  and that  $a \wedge b$  is not an anti-chain of  $a \wedge c$ , and not that  $b \geq a$  or  $c \geq a, b$ . Although this case also allows that  $b \vee c \in \uparrow a \Rightarrow b \cup c \in \uparrow a$  does not hold, the procedure (vii) in definition 4 can be applied, so  $a$  is removed and a distributive sub lattice is then obtained, as shown in Fig. 8 below.

(iii) Another case is from (i) and (ii). Because the statement  $a \wedge (b \vee c) \geq (a \wedge b) \vee (a \wedge c)$  holds in any lattice, we will prove that  $a \wedge (b \vee c) \leq (a \wedge b) \vee (a \wedge c)$ . Because  $a \wedge (b \vee c)$  is a lower bound for  $\{a, b \vee c\}$ , we obtain  $a \wedge (b \vee c) \leq a$  and  $a \wedge (b \vee c) \leq b \vee c$ . In a topological system, it means that

$$a \in \uparrow(a \wedge (b \vee c)) \text{ and } b \vee c \in \uparrow(a \wedge (b \vee c)). \quad (13)$$

Because of the changeability of disjunction and join, we can replace this statement with

$$a \in \uparrow(a \wedge (b \vee c)) \text{ and } (b \in \uparrow(a \wedge (b \vee c)) \text{ or } c \in \uparrow(a \wedge (b \vee c))). \quad (14)$$

Because a logical statement satisfies the distributive law, this equation can be rewritten as  $(a \in \uparrow(a \wedge (b \vee c)) \text{ and } b \in \uparrow(a \wedge (b \vee c))) \text{ or } (a \in \uparrow(a \wedge (b \vee c)) \text{ and } c \in \uparrow(a \wedge (b \vee c)))$ .

Additionally, due to the changeability of conjunction and meet, we obtain

$$a \wedge b \in \uparrow(a \wedge (b \vee c)) \text{ or } a \wedge c \in \uparrow(a \wedge (b \vee c)). \quad (15)$$

By replacing the disjunction with join,

$$(a \wedge b) \vee (a \wedge c) \in \uparrow(a \wedge (b \vee c)). \quad (16)$$

Thus, we finally obtain

$$a \wedge (b \vee c) \leq (a \wedge b) \vee (a \wedge c). \quad (17)$$

**Significance of Distributivity.** What is the significance of a distributive lattice? It is an abstract expression of a way of thinking in which anything can be considered as the result of summation, which is also known as a representation theorem for a distributive lattice (Davey and Priestley, 2002). The theorem states that any distributive lattice can be expressed as a lattice of sets consisting of sets equipped with a join that is defined by a union and meet, which are defined by an intersection.

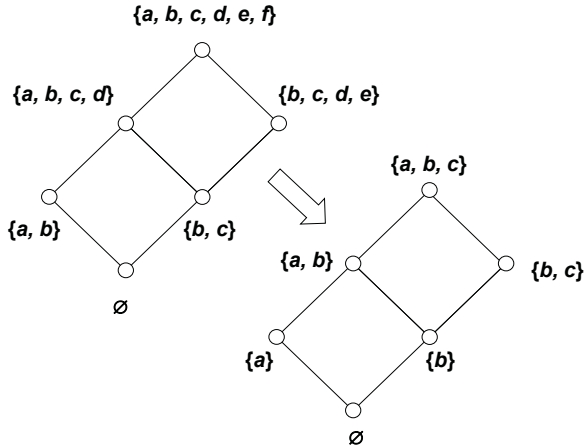


Figure 9. A distributive lattice (above) can be a lattice of sets (below) (i.e., a join and meet can be defined by a union and intersection, respectively).

Fig. 9 shows a representative example. The above Hasse diagram exemplifies a distributive lattice containing elements that are sets. Notice that join does not equal union and that meet does not equal intersection while meet is  $\{a, b\} \wedge \{b, c\} = \emptyset$  and intersection is  $\{a, b\} \otimes \{b, c\} = \{b\}$  in Fig. 9 (above). Similarly, while  $\{a, b\} \vee \{b, c\} = \{a, b, c, d\}$ ,  $\{a, b\} \oplus \{b, c\} = \{a, b, c\}$ , which explains why the lattice in Fig. 9 (above) is not a lattice of sets. However, this lattice can be represented by a set of lattices by replacing elements with other elements represented in Fig. 9 (below). In this lattice, any meet is defined by an intersection, and any join is defined by a union.

Thus, a distributive lattice is an abstract expression of set-based thinking: a whole system can be reduced to elements, and summing up the elements can create a whole system. There is no non-linear interaction among the elements. Our results, in which an evolutionary topological system evolves towards a distributive lattice, indicate that a cause-effect relationship in space-time can be developed as the simplest logical structure. Although the changeability of disjunction and union (conjunction and meet) can be erroneous, space-time appears to be constructed as an operationally simple cause-effect relationship.

### Out of body experience resulting from the changeability of disjunction and join

An evolutionary topological system is based on the changeability of join and disjunction and of meet and

conjunction. What is the real implementation of the changeability? Although a disjunction or conjunction is a type of distribution or a set of elements in a causal set, join and meet are single elements of a causal set. However, they differ with respect to logical status and can be replaced with each other to improve the conflict between Point and Open Logic. We believe that this changeability plays an essential role in our cognitive system.

**Body image.** The generation of a body image can be one of the examples resulting from the changeability of disjunction and join. In brain science, the relationship between body schema (operational body) and body image (body owned by oneself) is investigated. Even a hermit crab can detect a sudden change in the carried shell in terms of size, thereby changing its method of walking based on the shell size (Sonoda et al., 2013). This observation implies that body schema appears to affect body image and that both always interact with each other.

Although a body schema is based on controlling a point, a body image is a collection of parts as a whole. The former is related to Point Logic, and the latter is related to Open Logic. Thus, the interaction between body image and schema is also faced with the conflict between Point and Open Logic.

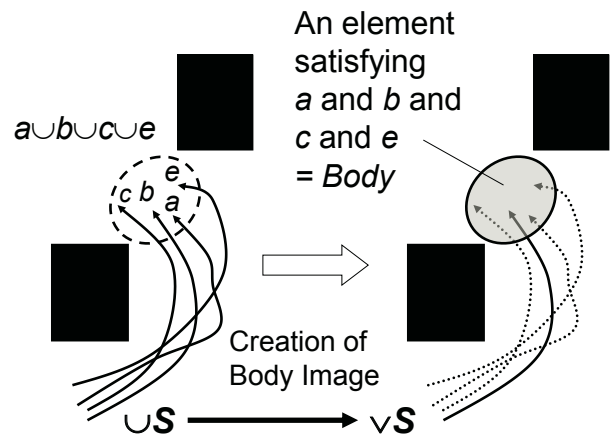


Figure 10. A model for the creation of a body image resulting from the changeability of disjunction and join. The black squares are pillars or obstacles.

Imagine an infant who operates his own body. He first assumes that he is just a point and has to control this point to walk between pillars in his room (Fig. 10 left). He never controls this point in a strict sense. However, he attempts to move this point to a central point between the pillars, and the point is occasionally at a location  $a$  and occasionally at  $b$  (because he ignores his body size). As a result, a body can be at  $a$  or  $b$ . However, if an idea occurs to him that the possibility of  $a$  or  $b$  itself can be a “big” point, then a point has to be something big that is at  $a$  and  $b$  at the same time (Fig. 10 right). Thus, the body image was created as a join resulting from disjunction.

**Out of body experience.** According to our evolutionary topological system, two events that are exclusive of each other can be replaced with a single event, which can satisfy two

exclusive events in a causal set. Thus, we can design an artificial space-time event appearing from the changeability of disjunction and join. Using a variation of the Substitutional Reality (SR) system (Suzuki et al., 2012), we construct the sensation of an out-of-body experience, as shown in Fig. 11.

The system consists of a head-mounted display (HMD) fitted with a video camera at the front center (subjective-eye camera), a panoramic video camera (objective-eye camera) and a control computer. In our preliminary experiment, a participant sitting in a room first sees an experimenter in front of him with his naked eyes. Then he wears the HMD and see the experimenter through the subjective-eye camera. This causal relationship is shown as an event,  $a \leq a'$  (Fig. 11). Then, the scene pre-recorded by the objective eye camera set in front of the participant is projected in the HMD. Thus, the participant sees himself appearing and wearing the HMD, corresponding a causal relationship of  $b \leq b'$  (Fig. 11). With several virtual-reality-inspired tricks, even in the objective view he is able to look around freely as in the subjective view.

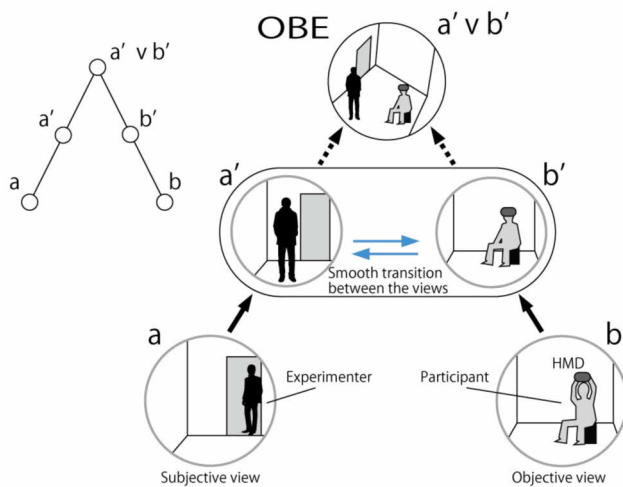


Figure 11. Design for the “Out-of-Body Experience (OBE)” in an evolutionary topological system.

The subjective view,  $a'$ , and the objective view,  $b'$ , are exclusive of each other, although they are both sides of the same coin –“now”. They are not united by a single event in this situation. However, if the subject experiences a smooth transition between objective and subjective view (by changing the objective camera position and using several video effects), represented by the blue arrows in Fig. 11, he feels as if he is seeing himself in his subjective view. That feeling corresponds to the replacement of  $a' \cup b'$  by a single event,  $a' \vee b'$ . According to his verbal report, he feels an OBE which is not just an experience of seeing himself. Instead, he feels as if he has created another perspective by his imagination (Fig. 11). Therefore, in this feeling, exclusive subjective and objective scenes can be considered to be united as a single event, which is different from the feeling experienced in a previous experiment (Lenggenhager et al., 2007).

## Conclusion

A causal set developed in quantum physics attempts to describe a space-time from an observer’s view. If so, we have to pay attention to the interaction between the causal relationship and an observer that is a computation and/or logical operation in a space-time equipped with a causal relationship. For this purpose, we describe a causal set and its semantics as a topological system consisting of Point and Open logic.

The binary relationship between Point and Open Logic can derive a conflict that can be improved by restricting logical operations. We here, however, propose an evolutionary topological system in which a conflict between Point and Open Logic is locally and temporally improved that can generate an artificial causal relationship. This local improvement is implemented by the changeability between join and disjunction and between meet and conjunction represented by the replacement of a set with an element to keep non-restricted logical operations.

We show and verify that a causal set of the evolutionary topological system can converge to a distributive lattice that is an abstract expression of the simplest logical operation for summation. We also show that the changeability of disjunction and join can generate abnormal space-time feelings, such as an out-of-body experience. We can design both normal and abnormal artificial space-time based on an evolutionary topological system.

## References

- Bombelli, L. Lee, J. Meyer, D. and Sorkin, R.D. (1987) Space-time as a causal set. *Phys.Rev.Lett.* 59: 521-524.
- Davey, B.A. and Priestley, H.A. (2002) *Introduction to Lattices and Order*. Cambridge Univ. Press.
- Ehrsson, H.H., Spence, C., & Passingham, R.E. (2004). That’s my hand! Activity in premotor cortex reflects feeling of ownership of a limb. *Science*, 305: 875-877.
- Grey, W. (1997) Time and becoming. *Cogito* 11(3), 215-220.
- Gunji, Y.-P., Haruna, T., Uragami, D. and Nishikawa, A. (2009), Subjective spacetime derived from a causal histories approach.. *Physica D* 238: 2016-2023.
- Klugry, A.L. Sepanina, I.V. (2011) An example of the stochastic dynamics of a causal set. *arXiv:1111.5474v1 [gr-qc]* 2011.
- Lenggenhager, B., Tadi, T., Metzinger, T., & Blanke, O. (2007). Video ergo sum: manipulating bodily self-consciousness. *Science*, 317: 1096-1099.
- Markopoulou, F. (2000) The internal description of a causal set: what the universe looks like from the inside. *Comm. Math. Phys.* 211: 559-583.
- McTaggart, J.M.E. (1908) The unreality of time. *Mind* 17(68), 45-74.
- Mellor, D.H. (1998) *Real Time II*, Routledge.
- Scott, D. S. (1976) Data type as lattice. *SIAM J. Comput.* 5(3): 522-587.
- Sonoda, K., Asakura, A., Minoura, M., Elwood, R.W. and Gunji, Y.P. (2012) Hermit crabs perceive the extent of their virtual bodies. *Biol Lett.* 8(4): 495-497.
- Suzuki, K., Wakisaka, S. and Fujii N. (2012) Substitutional reality system: A novel experimental platform for experiencing alternative reality. *Sci. Rep.* 2: 459.
- Vickers, S. (1996) *Topology via Logic*. Cambridge Univ. Press.

## "Artificial spores" - hybrid alginate microcapsules with encapsulated yeast cells

Jitka Čejková, Petra Haufová, Damian Gorný and František Štěpánek

Chemical Robotics Laboratory, Institute of Chemical Technology Prague, Technická 3, 166 28 Prague 6, Czech Republic  
Jitka.Cejkova@vscht.cz

### Abstract

The present work describes the fabrication, structure and functional characterization of composite microcapsules containing encapsulated viable yeast cells, fluorescently labeled liposomes and magnetic nanoparticles embedded in a calcium alginate matrix produced by ink-jet printing. The proliferation of the encapsulated cells under favorable conditions (presence of nutrients, temperature) is used as a biological trigger for the disintegration of the microcapsules and the liberation of the encapsulated sub-micron particles. The principle of "artificial spores", *i.e.* the possibility to repeatedly stop and restart the cell proliferation process has also been demonstrated. Such biologically triggered release from composite microcapsules is novel and offers potentially interesting applications such as autonomous release of bactericides or fungicides only under conditions that are favorable for microbial growth.

### Introduction

In biology, a spore is defined as a reproductive structure that is adapted for dispersion and survival for extended periods of time in unfavorable conditions. Once conditions are favorable, spores can develop into new organisms. The activators of such a transformation from spore to cell could be *e.g.* nutrients, temperature, pH, or combination of these parameters. There are several reasons why the spores are formed in the nature:

- (i) Spores allow the organisms to survive for many (in some cases, millions of) years under adverse conditions, thus they serve as storage system for genetic information.
- (ii) Spores shield cellular components in harsh conditions, and so spores have the protection function.
- (iii) A spore must somehow "arrive" at a location and be there at a time favorable for germination and growth. Some spores have flagella or other organelles that serve for the species dispersion to longer distances and new areas. Therefore spores serve as transporters of genetic information.

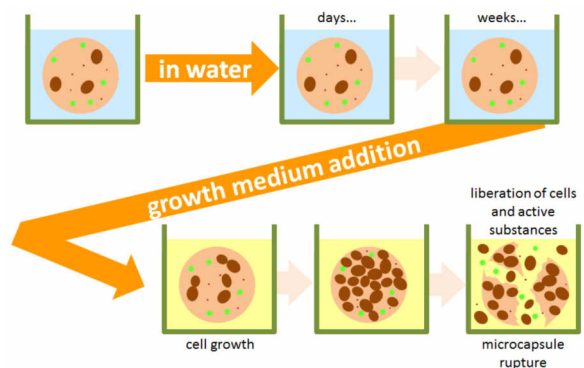
In present work we adopted the idea of spores and created hybrid alginate microcapsules with embedded yeast cells that are long-term stable and inactive and perform specific target mission only after activation by changing the conditions in their surround. The target aim is to disintegrate and liberate and disperse the encapsulated content in proper time. The principle of such artificial spores is described in Figure 1. Under unfavorable conditions (absence of nutrients) no cell division of embedded yeast occurs and the composite microcapsules are stable in aqueous medium for extended periods of time without disintegration or release of their

content. Once the microcapsules encounter favorable conditions (presence of nutrients, here provided by a culture medium), cell division and growth causes a rupture of the alginate capsule and release of the embedded components. Liposomes loaded with fluorescein represent a model "active" particulate substance that is to be liberated from the composite microcapsules. Additionally, iron oxide magnetic nanoparticles were also embedded within the composite microcapsules to facilitate their manipulation and separation by a magnetic field.

The present paper focuses mainly on problem (i) described above. Artificial spores were fabricated and their stability and inactivity for long time were investigated. The ability to activate in suitable conditions (in this case nutrients additions) was studied. Artificial spores cultivated in growth medium showed the ability to disintegrate and release embedded object into the surround. This mechanism acts as a biological trigger for controlled opening of the microcapsule.

Further we concentrated on the protection function of spores (task (ii) above). Artificial spores were coated by solid silica shell and the viability of encapsulate yeast was tested. Although the coating process does not kill the cells, the cell growth in the microcapsules was not sufficient for the microcapsule disruption. Unfortunately this way of protection shell formation seems to be unsuitable.

Such artificial spores will find applications in biologically triggered controlled delivery *e.g.* of natural fragrances or benign fungicides. Another application of these objects could be as intelligent indicators of storage quality.



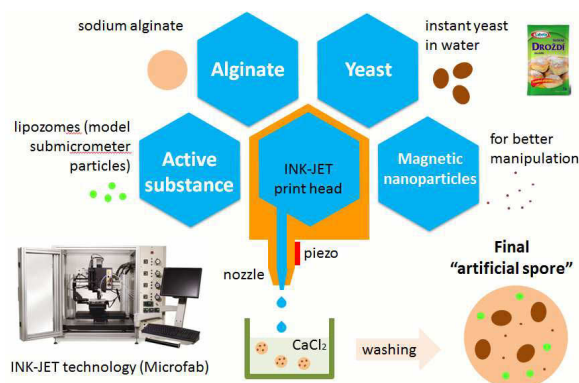
**Figure 1:** Schematic principle of artificial spore rupture and liberation of an active substance into the environment caused by yeast cell growth in the culture medium.



## Experimental

### Materials

Sodium alginate, calcium chloride ( $\text{CaCl}_2$ ), (3-aminopropyl) trimethoxysilane (AMPS), tetramethoxysilane (TMOS), *n*-hexane, [7 - diethylaminocoumarin - 3 - carboxylic acid] (DEAC), fluorescein diacetate (FDA), yeast extract and glucose were purchased from Sigma-Aldrich. Instant yeast cells (Labeta a.s., Czech Republic) were suspended in deionized water in various concentrations (1 mg of dry powder corresponds to  $3 \times 10^7$  cells). Hydrophilic iron oxide nanoparticles were prepared according to a synthesis described in (Tokárová et al., 2012). Fluorescently labeled liposomes (molar ratio of DPPC : cholesterol was 2:1) were synthesized in the same way as described in (Ullrich et al., 2013). Deionized water was produced by a ionex filter (Aqual 25).



**Figure 2:** Schema of artificial spore preparation. Solution of sodium alginate, yeast cells, liposomes and magnetic nanoparticles is by means of Ink-Jet printing technology dropped into  $\text{CaCl}_2$  solution. Precipitated calcium alginate microcapsules are subsequently washed.

### Artificial spore preparation

All artificial spores were produced by inkjet printing (Dohnal and Štěpánek, 2010). A piezoelectric drop-on-demand print-head type M5-ABP-01-80-6MX supplied by Microfab, Inc. (Plano, Texas, USA) was used, coupled with a control unit type JetDrive III and a pressure controller type CT-PT-01 also supplied by Microfab, Inc. 2 ml of aqueous solutions of 2% (w/w) sodium alginate and 2 ml of aqueous suspension of yeast cells were mixed and printed into approximately 50 ml of aqueous solution of 2% (w/w)  $\text{CaCl}_2$  where a rapid ionic cross-linking of the microdroplets occurred. The receiving  $\text{CaCl}_2$  solution was constantly agitated to avoid microdroplet coalescence after impact. To prepare the magnetic microcapsules, one half of the cell suspension was replaced by citrate-stabilized iron oxide nanoparticle dispersion in water (15 mg/ml). The solution for printing of magnetic capsules containing liposomes was mixed from a sodium alginate solution, the cell suspension, the iron oxide nanoparticle solution and a liposome solution in the volume

ratio 4:1:1:2. Cross-linked calcium alginate microcapsules were separated from the  $\text{CaCl}_2$  solution by using filter or magnet and suspended in deionized water in which they were stored at room temperature until further use. In this state the composite microcapsules were stable for up to 4 months without any significant loss of yeast cell viability or leakage from the liposomes.

### Coating of artificial spores by silica shell

The silica shell was formed by a sol-gel process according to our previous procedure (Haufova et al., 2012) derived from the work of Sakai (Sakai et al., 2001): the alginate microcapsules were suspended in *n*-hexane and kept at  $4^\circ\text{C}$  in an ice-bath. AMPS and TMOS were then added to *n*-hexane containing the alginate particles. AMPS was added first and stirred for 1 min, followed by TMOS and stirring for another 1 min. The thickness of the silica layer is influenced by the quantity of the silica precursors (AMPS and TMOS). The volume ratios of 10:14:0.8:0.6 for alginate:*n*-hexane:AMPS:TMOS were used in our case. Based on the assumption of complete hydrolysis of the alkoxysilanes and average alginate particle size of  $70\ \mu\text{m}$ , the resulting thickness of the deposited silica layer is  $0.23\ \mu\text{m}$ . The resulting microparticles were then rinse with 1.0 wt.%  $\text{CaCl}_2$  solution and then kept in deionized water in a fridge for further use. All the procedures/solutions were kept cold (at  $4^\circ\text{C}$ ) to enhance the stability of liposomes.

The synthesis of fluorescently labeled silica nanoparticles is specified in (Čejková et al., 2010) with the exception that the fluorescent dye DEAC was used instead of fluorescein isothiocyanate (FITC). For subsequent visualization of the silica layer, the pre-synthesized fluorescently labeled silica nanoparticles ( $\text{SiO}_2$ -DEAC nano) were added into the silica layer formed by the sol-gel process. 100 mg of  $\text{SiO}_2$ -DEAC nanoparticles was mixed with 1 ml AMPS for 24 hours prior the further sol-gel procedure. The mean diameter of the  $\text{SiO}_2$ -DEAC nanoparticles was 150 nm.

### Artificial spore characterization

The artificial spores were characterized by means of inverted optical microscope (Olympus CK40) and a laser scanning confocal microscope - LSCM (Olympus Fluoview FV1000). The particle size was evaluated by laser diffraction (Horiba Partica LA 950/V2). The viability of yeast cells was confirmed by using the standard fluorescein diacetate solution method.

### Yeast cell division and artificial spore disruption study

For a study of the cell division and disintegration of artificial spores, the composite microcapsules were placed into a Petri dish containing a culture medium (consisting of glucose in a concentration of 10 g/l and yeast extract in a concentration of 5 g/l) and monitored by an optical microscope for 24 hours. The cell growth curves were measured by means of visible spectrophotometer (Specord 205 BU, Analytik Jena, Germany); the wavelength used for the measurement of optical density was 600 nm ( $\text{OD}_{600}$ ).

## Results and discussion

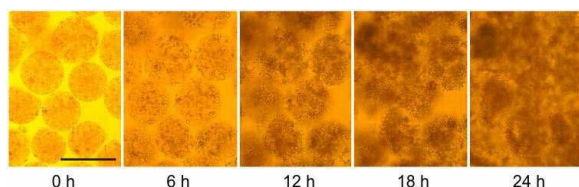
### Artificial spores characterization

The drop-on-demand inkjet technology was used for the formation of calcium alginate microcapsules with embedded yeast cells by ejecting droplets of a sodium alginate precursor into a pool of calcium chloride solution. The shape of the formed microcapsules was mostly spherical, however, some of them were distorted (flattened) due to droplet deformation upon landing into the  $\text{CaCl}_2$  solution.

The viability of yeast cells in the composite microcapsules was confirmed by using fluorescein diacetate (FDA). This colorless compound exhibits no fluorescence, however, it is known that it diffuses through cell membrane and living cells are able to hydrolyze it by their enzymatic apparatus and transform FDA into fluorescein (Adam and Duncan, 2001). Typically, 2 ml of microcapsule suspension were incubated with a few droplets of FDA in acetone for 20 minutes, then washed and observed under LSCM. By this test it was proven that the cells are able to retain their viability during the ink-jet printing process. The microcapsules after one month of their fabrication and storage in water that were incubated with FDA show in LSCM images green spots. These spots correspond to living cells, which confirms that the cell viability is preserved for many weeks.

The hybrid microcapsules with embedded yeast cells were stored in water for few weeks and no microcapsule changes and cell division in capsules were observed. Radical changes occurred after incubation with growth medium containing yeast extract and glucose.

To observe the cell division of encapsulated yeast, composite microcapsules were suspended in Petri dish with growth medium and placed under microscope. Images in 1-minutes intervals were grabbed for at least 24 hours. Typical results are shown in Figure 3 for microcapsules containing yeast cells in concentration of  $3.75 \times 10^8$  cell/ml.



**Figure 3:** Yeast cell division in alginate microcapsules incubated in a Petri dish with culture medium. Concentration of yeast cells in prepared microcapsules  $3.75 \times 10^8$  cell/ml. Scale bar represent 100  $\mu\text{m}$ .

Figure 3 shows the changes of microcapsules in various times in static conditions in a Petri dish with cultivation medium. After fabrication the cell concentration in microcapsules corresponded to the cell concentration in alginate matrix that was printed into  $\text{CaCl}_2$  solution. Time  $t = 0$  h corresponds to placing of microcapsules into cultivation medium. Few hours after incubation in growth medium, buds on cells appeared and cells started to divide. Around the time  $t = 7$  h, the cell division was highly developed. Around the time  $t = 15$  h, alginate microcapsules

were almost full of cells and at time  $t = 18$  h first ruptures of composite microcapsules started. Due to intensive cell division all microcapsules disintegrate and almost no compact round microcapsules were evident at time  $t = 24$  h. In the Petri dish only clusters of yeast cells were present. This experiment confirmed the idea of artificial spore – microcapsules are for long time stable and after external condition changes ruptured and released their content.

Further the growth curves of cells in alginate microcapsules were measured by using visible spectrophotometry as a function of optical density at the wavelength 600 nm. First few hours no changes were evident, because yeast cells were still in lag phase of their growth, they adapted themselves to growth conditions and rarely cells divided. About at the time  $t = 7$  h, the cells entered into the exponential phase of their growth which corresponded with rapid increase in measured absorbance. Approximately until the time  $t = 24$  h the increase of absorbance is evident, which shows the cell division. After the time  $t = 24$  h, the absorbance did not increase, because cells entered into the stationary phase of their growth due to the lack of nutrients.

As was shown above, composite microcapsules are stable in water, whereas after cultivation in growth medium they are able to disintegrate and release their content. To confirm this fact, the liberation of fluorescently labeled liposomes was observed by laser scanning confocal microscope. Directly after their fabrication, the fluorescence signal is obtained only in microcapsules and this confirms the successful encapsulation of liposomes into microcapsules. The same microcapsules imaged one day after fabrication and storage in pure water show the fluorescence signal again only in microcapsules, no liberation of liposomes from microcapsules occurred. On the other hand, after cultivation of microcapsules in growth medium, the cells divide and cause the rupture of capsules, the liberation of all encapsulated substances and their release into the surround. After one day of incubation in culture medium no compact microcapsules were present, only the clusters of cells were evident and the fluorescence signal was detectable from whole medium because the liposomes were released during the microcapsule disintegration.

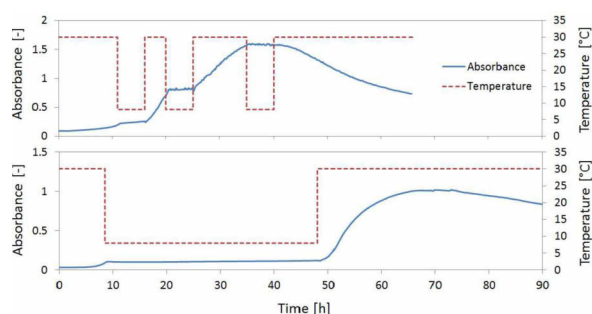
### Temperature effects on artificial spore germination

In biology, a spore is defined as a reproductive structure that is adapted for dispersion and survival for extended periods of time in unfavorable conditions. The interesting property of the transformation of spores into cellular entities is as follows: once the conditions seem to be suitable for germination, the spores enter a lag phase and activate specific genes that trigger certain signal pathways leading to swelling and cell emergence. Once a spore has swollen, germination becomes irreversible, but during the lag phase activated spores can return to dormancy (Van Dijken and Van Haastert, 2001).

To cover the idea of “artificial spores”, additional experiments with switching favorable/unfavorable conditions were performed. In previous section the experiments with nutrient additions were described. It was shown, that the growth medium can start the process of cell division with consequent microcapsule rupture. We were interested if it is

possible to stop the division process anyway and then to trigger the growth again. Therefore following experiments with temperature changes were done.

The experiment mimicking spore response to conditions changes was performed in spectrophotometer equipped with heating/cooling facility. The growth curves were measured and with temperature changes compared. The experiment started at unfavorable conditions, where microcapsules were stored in water (without nutrients). The first change consisted in placing of capsules in the culture medium at temperature 30°C (corresponding to time  $t = 0$  h in the growth curves in Figure 4). Although they had enough of nutrients and the temperature was acceptable, the first few hours the yeast embedded in microcapsule were still dormant, because they were in the lag phase. When they entered the exponentially phase of their growth, we have decreased the temperature rapidly to 8°C. Such a temperature shock stopped the cell growth resulting in a return into dormancy of microcapsules. During this time no change of optical density was observed corresponding to no division of cells. This horizontal line in the growth curve confirmed the possibility to paralyze artificial spores with temperature decrease that represents coming back of unfavorable conditions. To provide the better conditions again, the temperature was increased back to 30°C. Such a sudden suitable condition establishment lead to rapid cell growth again. The next temperature decrease caused the next interruption of cell division. The example of three repetitions heating/cooling represents Figure 4(B) shows the possibility to “freeze” the cell growth in microcapsule for almost two days.



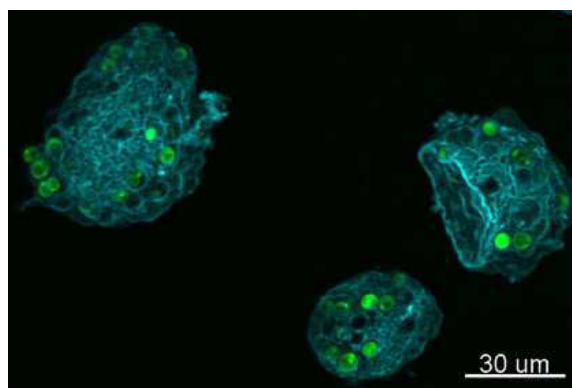
**Figure 4:** Interruption of yeast growth by temperature decrease and resumption of growth by increase of temperature. (A) Three interruption of cell growth by three temperature decreases (at time 11-16 h, 20-25 h, 35-40 h). (B) Long cell growth interruption for almost two days (8-48 h).

### Silica shell of artificial spores

To develop the concept of artificial spore it is necessary to develop the strategy to coat the particle with any hard thin shell that could protect the encapsulated content. For this reason we decided to cover the hybrid alginate microcapsule by silica shell because of our previous experience with this process (Haufová et al., 2010).

Figure 5 represents image of microcapsules covered by thin fluorescently labeled silica layer (it is visualized by blue color). Green spots correspond to viable encapsulated yeast cells visualized by using FDA as described in previous

sections. It was proven that the sol-gel coating process in harsh conditions (temperature 4°C, in *n*-hexane solution) does not kill all the cells and some cells are still viable. It can be concluded that at least some of the yeast cells were able to survive both the inkjet printing and the silica coating processes, although there seems to be also a fraction of dead (non-fluorescent) cells. However the coating lead to microcapsule shrinkage (*ca* to 30 µm compared to 80 µm original diameter).



**Figure 5:** Laser scanning confocal microscopic images of alginate microcapsules with encapsulated yeast cells (viable cells visualized by green color) and covered by fluorescently labeled silica shell (blue color).

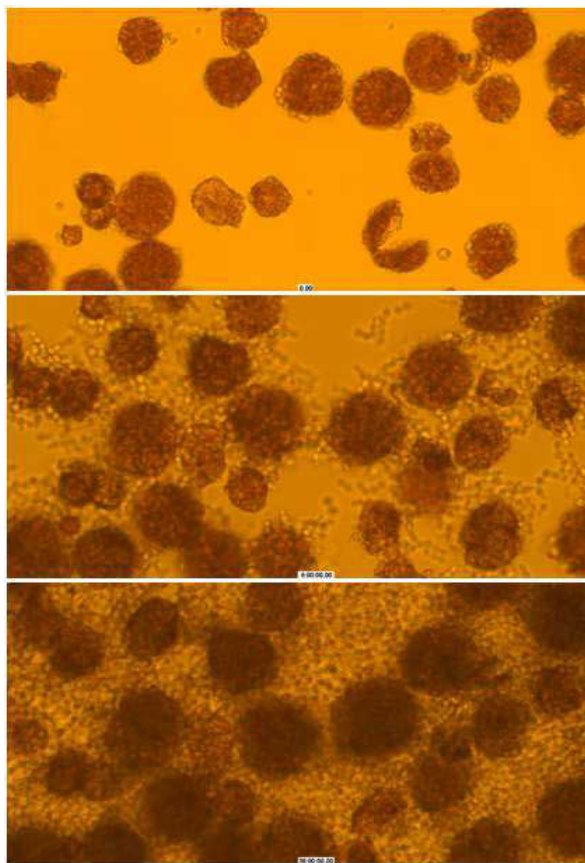
Further we focused on cell division in silica-coated microcapsules. Figure 6 summarized the results of silica-coated artificial spores' cultivation in growth medium. It is evident, that yeast cells entrapped on the surface of microcapsules are able to divide and growth freely in surrounding medium, however the growth of cells inside the microcapsule is not sufficient to microcapsule disruption and following rupture. Unfortunately these results exclude the sol-gel silica coating method as a suitable process for protecting surface layer formation. Our future work will focus on other cover techniques, such as layer-by-layer method.

### Dry artificial spores

All experiments described above were performed in aqueous conditions. Artificial spores were fabricated and stored in aqueous solutions and never during their studies were dried. Following test focused on the properties changes after particle drying.

The artificial spores were air dried and Figure 7 represents microscopic images of their rehydration. In (A) dry shrunken microcapsules are displayed. Figure (B) shows the result of microcapsule incubation in water for 1 hour. Dry alginate is not able to swell in pure water and microcapsule retain in the shrunken state. On the other hand, growth medium containing various ions is able to facilitate the microcapsule swelling (Figure C). Unfortunately, cell division in rehydrated artificial spores was not sufficient for microcapsule rupture. This observation recommends the artificial cells applications in hydrated state without any drying step.



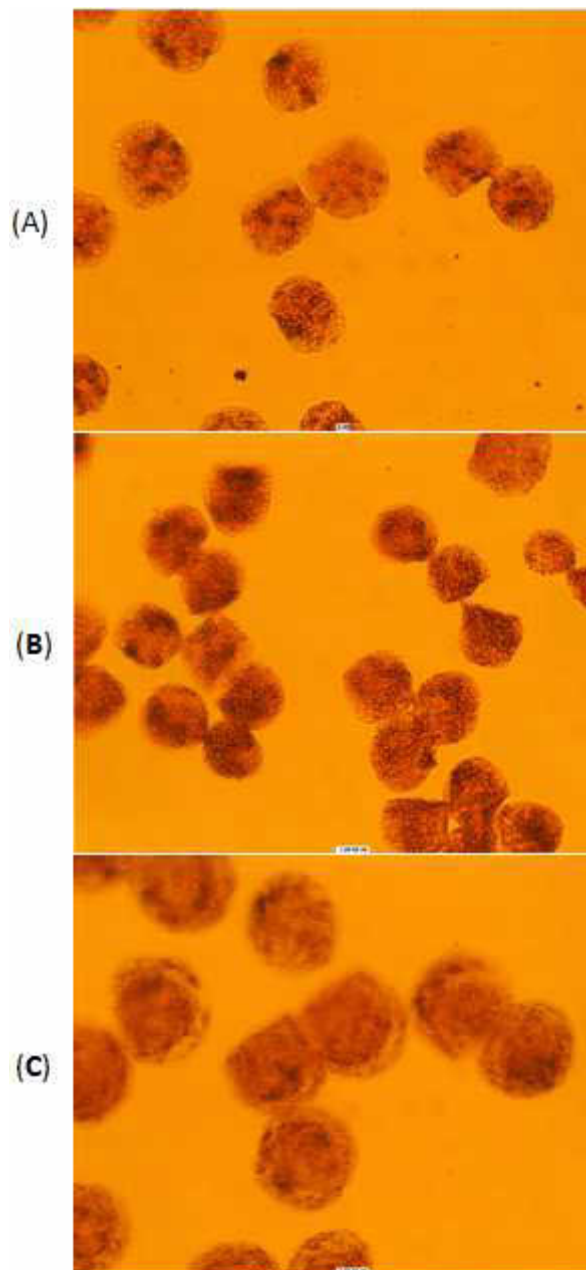


**Figure 6:** Cultivation of silica-coated artificial spores in growth medium. Optical microscope images from times  $t = 0$ , 6 and 36 hours after placing into the medium.

### Conclusions

We fabricated a new type of microcapsules with embedded microorganisms that can locally liberate sub-micrometer objects, and act as a biological trigger for controlled opening of the microcapsule. Such a hybrid microparticle cover the idea of artificial spore, that is inactive for long time without any changes and start to rupture after external condition changes (here nutrient addition). Artificial spores prepared here were able to liberate encapsulate model substance in form of liposomes. It was shown, that silica coating by means of so-gel process does not allow the microcapsule rupture as in the case without any shell presence. Further it was observed, that also dried artificial spores are not able to rupture after rehydration in cultivation medium.

Our future work will focus on microcapsule covering to form a shell that will protect the alginate and encapsulated objects and subsequently will not prohibit the particle rupture. Because the rupture of capsules is also impossible for particles that were dried, we will concentrate to find applications for artificial spores in aqueous solution, such as in biologically triggered controlled delivery, *e.g.* of natural fragrances or benign fungicides or as quality indicators.



**Figure 7:** Microscopic images of rehydration of dry artificial spores (A) in water (B) and in growth medium (C) after 1 hour.

**Acknowledgments.** Thanks Jaroslav Hanuš for fluorescently labeled liposomes. Financial support from the European Research Council (grant number 200580-Chobotix) is gratefully acknowledged.



## References

- Adam, G., and Duncan, H. (2001). Development of a sensitive and rapid method for the measurement of total microbial activity using fluorescein diacetate (FDA) in a range of soils. *Soil Biology and Biochemistry*, 33: 943-951.
- Čejková, J., Hanuš, J., and Štěpánek, F. (2010). Investigation of internal microstructure and thermo-responsive properties of composite PNIPAM/silica microcapsules. *Journal of Colloid and Interface Science*, 346: 352-360.
- Dohnal, J., and Štěpánek, F. (2010). Inkjet fabrication and characterization of calcium alginate microcapsules. *Powder Technology*, 200: 254-259.
- Haufová, P., Dohnal, J., Hanuš, J., and Štěpánek, F. (2012). Towards the inkjet fabrication of artificial cells. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 410: 52-58.
- Sakai, S., Ono, T., Ijima, H., Kawakami, K. (2001). Synthesis and transport characterization of alginate/aminopropyl-silicate/alginate microcapsule: application to bioartificial pancreas. *Biomaterials*, 22: 2827-2834.
- Tokárová, V., Pittermannová, A., Čech, J., Ulbrich, P., and Štěpánek, F. (2012). Thermo-responsive adhesion properties of composite hydrogel microcapsules. *Soft Matter*, 8: 1087-1095.
- Ulrich, M., Hanuš, J., Dohnal, J., and Štěpánek, F. (2013). Encapsulation stability and temperature-dependent release kinetics from hydrogel-immobilised liposomes. *Journal of Colloid and Interface Science*, 394: 380-385.
- Van Dijken, P., and Van Haastert, P. (2001). Phospholipase Cdelta regulates germination of *Dictyostelium* spores. *BMC Cell Biology*, 2: 25.

## ***In vitro* Synthesis of Membrane Protein Machinery toward the Construction of Artificial Cell**

Hideaki Matsubayashi<sup>1</sup>, Yutetsu Kuruma<sup>1</sup> and Takuya Ueda<sup>1</sup>

<sup>1</sup>Department of Medical Genome Sciences, Graduate School of Frontier Sciences,  
The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa-shi, Chiba, Japan  
kk116432@mgs.k.u-tokyo.ac.jp

### **Abstract**

Cell membranes are essential part of living cells. They are important as the envelope which encapsulate the biochemical systems within them and distinguish “self” components from “not-self” surrounding environment. Furthermore, cell membranes function as an interface which exchange materials between inside and outside of the cell, sense external environment, and transmit signals to inside systems to response the circumstance. In the field of synthetic biology, liposome (lipid membrane vesicle) has been widely used as a model of cell membrane. Although liposome is certainly a good model as for cell envelope, it is not satisfying the biochemical functions of cell membranes. Since the most of cell membrane functions are led by membrane embedded proteins, we should combine membrane proteins with liposome to construct more feasible artificial cell membranes. In this research, we aim to equip membrane machinery on liposome membrane using in vitro gene expression system.

We show that a membrane machinery Sec translocon<sup>1</sup>, which conducts membrane secretion and insertion of protein (Figure 1), has been synthesized onto the liposome membrane from its template DNAs. The gene expression was performed with the cell-free protein synthesis system, PURE system<sup>2</sup>. The PURE system is a reconstructed transcription/translation system that actualizes the phenomenon of Central Dogma (DNA-RNA-protein) in vitro with the minimal number of factors. Synthesized Sec proteins spontaneously localized at lipid bilayer and about 80nM Sec translocon were produced in functional state. This indicates that 2-3 Sec translocon were allocated to one liposome membrane based on a sequence of statistical calculations. Although the population density of the produced Sec translocon was not so high, a substantial peptides secretion activity of the Sec translocon was detected by biochemical assays. The specific activity of the synthesized Sec translocon was comparable to that of native Sec translocon isolated from cells in the function of protein secretion. In addition, the synthesized Sec translocon was able to conduct membrane insertion of a multi-spanning protein. These results indicate that the artificially synthesized Sec translocon is functional both in secretion and insertion. It should be noted that the formation of Sec translocon was achieved in self-assembly process.

Our results demonstrate that the functional Sec translocon has been constructed in totally synthetic manner. Although the

protein synthesis in this study were performed on the outside of the liposomes, the same reaction would be occurred inside liposomes, for instance giant unilamellar vesicles that can effectively encapsulate a cell-free system and DNA<sup>3</sup>. More importantly, our results raise a possibility that various membrane proteins can be subsequently produced in liposome membrane by primarily constructed Sec translocon, and eventually non-functional liposomes will gain divers bio-functions that are essential for a living artificial cell.

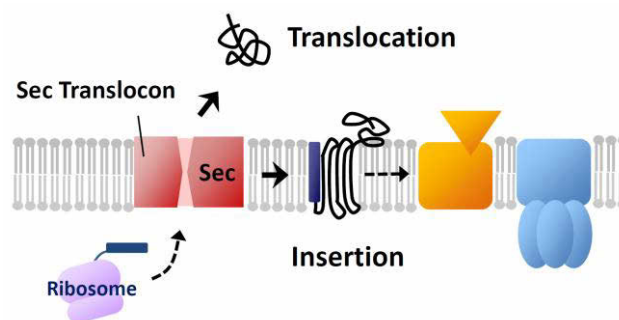


Figure 1: Sec translocon mediates translocation of secretory proteins and insertion of membrane proteins.

### **References**

1. du Plessis DJ, Nouwen N & Driessen AJ (2011) The sec translocase. *Biochim Biophys Acta* 1808(3): 851-8652.
2. Shimizu Y, *et al* (2001) Cell-free translation reconstituted with purified components. *Nat Biotechnol* 19(8): 751-755.
3. Stano, P, *et al* (2011) Compartmentalized reactions as a case of soft-matter biotechnology: synthesis of proteins and nucleic acids inside lipid vesicles. *Journal of Materials Chemistry* 21: 18887-18902.

## Experimental studied on growing chemical organisms.

Jerzy Maselko<sup>1</sup>, James Pantaleone<sup>2</sup>, Vitaliy Kaminker<sup>2</sup>

1. Chemistry Department, University of Alaska, Anchorage, USA
2. Department of Physics/Astronomy, University of Alaska, Anchorage, USA

The subject of this conference is: ‘Attempts to design and build artificial systems that display properties of organisms’. Two hundred years ago, the philosopher Immanuel Kant wrote the following:

‘Property of life: A self-propagating organization of processes.’

The difference between biological organisms and the complex chemical systems made by humans cannot be underestimated. The former contain such a large number of physical and chemical processes, each marked by incredible spatial and temporal organization and preciseness, that as of yet, their artificial reproduction is unachievable.

There remains, in addition, a gulf between biological growth and human controlled technology. These methods are not compatible. Human built complex chemical systems are assembled, whereas the biological systems are grown. Even the simplest biological cell cannot be disassembled and later reconstructed as if it were an AK-47.

Few known phenomena show promise of bridging this gulf; one of these is the ‘Chemical Garden’. In such systems, chemical reactions between a few elements drive fluid flow to spontaneously form precipitation structures. These structures can be grown from a ‘seed’, and the specific structure produced closely correlates to the composition of the seed and the environment. Chemical gardens have growth trajectories that span a vast morphological space, which includes hierarchical structures and also structures that move (chemical motors).

Among the first works devoted to these systems was published by Leduc in 1911 under the title “The Mechanism of Life”<sup>1</sup>. Leduc recognized the similarity of chemical gardens with biological systems and believed that this similarity could teach us something about the origin of life. In his book, he wrote that “The study of synthetic biology is therefore the study of physical forces and conditions which can produce cavities surrounded by osmotic membranes... and specialized their functions of living beings.”

Examples of the structures that can be grown in this manner are presented below.

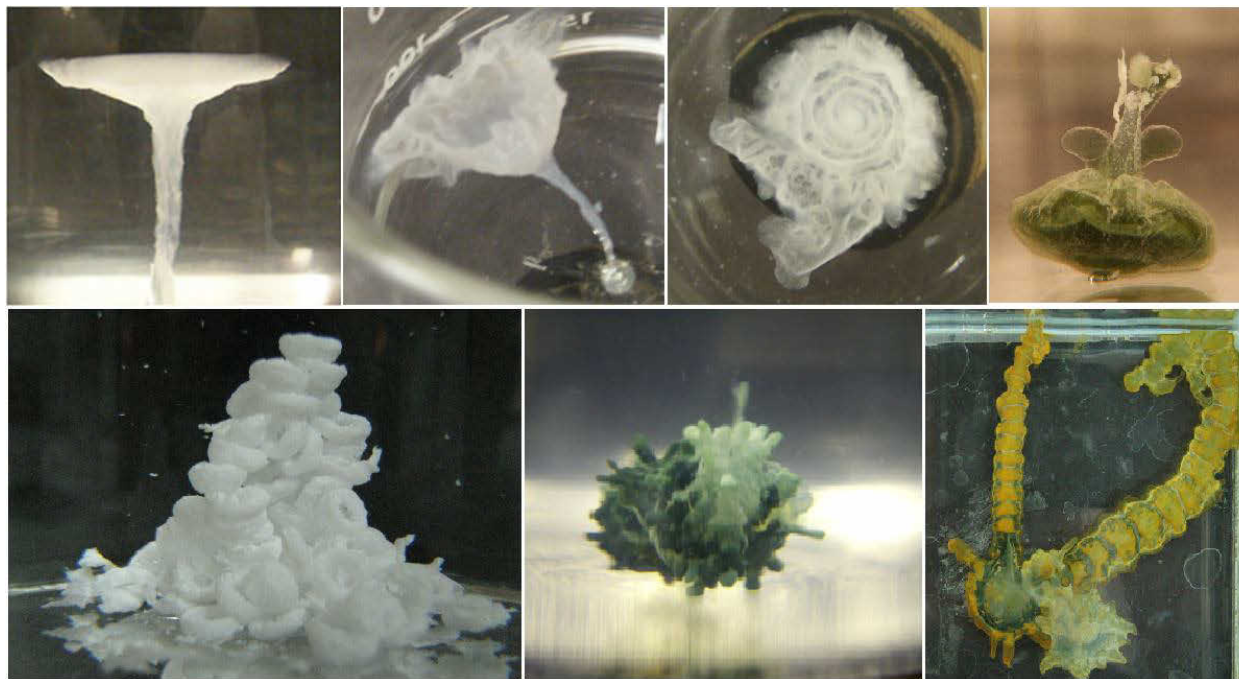


Figure1: Examples of complex structures that grow in simple chemical systems.

In each of these systems, growth is controlled by a complicated network of precisely organized physical and chemical processes. This growth is caused by a chain reaction; different potential gradients form structures that are, in turn, sources of new potential gradients. Thus we have a sequence of structures and processes: Structure follows process, gradient follows process, structure follows gradient, and so on. In these complex networks, different trajectories lead to different structures. Trajectories can be controlled by environment or seed composition. The growing process is often characterized by formation of templates that control formation of the next structure. The template may, depending on the trajectories, form different structures. Some structures may undergo metamorphosis where the entire structure changes (copper-oxalate system).

The process of growth is hierarchical, forming a network. Simple elements at lower levels form more complex structures at the higher levels. These, in turn, function as simple building blocks for even higher levels. So far, we have grown chemical systems with eight hierarchy levels. This is only the beginning. By changing system parameters, we can form different chemical building blocks, thus changing the trajectory. Trajectories may also be changed by catalysts or inhibitors (aluminum silicate system).

In most cases, the growing network can be divided into three parts: the seed, the construction, and the final structure. The seed forms an initial cell, where the initial potential gradients are making initial processes. The next part of network is construction, where the cascade of different structures are built. These systems are open ended, making it impossible to predict the final structure. Usually, the structure is finalized when the network has a loop that results in termination of structure growth.

Sometimes we observe the formation of a whole that may perform certain tasks. This part of the network is operational. It controls the task. Examples are cells that move up and down. This process is controlled by many chemical and physical sub-processes forming a loop (aluminum-carbonate-hydroxide system). Under another condition, this system may construct more complex structures that remain, made by humans, complex curtains.

The Chemical Garden and related structures are chemical systems bridging living and nonliving matter. They allow us to study a much simpler analogue to biological systems. The difference is the lack of DNA and genetic-informational systems that have very complex functions and control mechanisms. It may be said that chemical organisms are biological systems where DNA has been removed after formation of all proteins.

Mastering the growth of these complex chemical organisms may be the beginning of a new technology.

The following papers describe the presented phenomena:

Maselko, J., and P. Strizhak. 2004. Spontaneous formation of cellular chemical system that sustains itself far from thermodynamic equilibrium. *Journal of Physical Chemistry B*, 108, 4937 - 4939, doi:10.1021/jp036417j (2004).

Vladimir V. Udovichenko<sup>1</sup>, Peter E. Strizhak<sup>1</sup>, Agata Toth<sup>2</sup>, Dezso Horvath<sup>2</sup>, Steven Ning<sup>3</sup>, J. Maselko<sup>3,\*</sup> Temporal and Spatial Organization of Chemical and Hydrodynamic Processes. The system  $\text{Pb}^{2+}$  - Chlorite - Thiourea Accepted *J. Phys. Chem. A*, March 2008

A. Baker, A. Toth, D. Horvath, J. Walkush, A. Ali, W. Morgan, A. Kukovecz, J. Pantaleone, J. Maselko . Precipitation Pattern Formation in the Copper(II) Oxalate System with Gravity Flow and Axial Symmetry *J. Phys. Chem. A*, **2009**, 113 (29), pp 8243–8248

J. Pantaleone, A. Toth, D. Horvath, L. RoseFigur, J. Maselko, Pressure oscillations in Chemical Garden, *Phys. Rev. E* 79,056221 2009

A. Toth, D. Horvath, A. Kukovecz, A. Baker, S. Ali, J. Maselko “ Control of precipitation patterns formation in system Copper – Oxalate. *Journal of Systems Chemistry* 2012, **3**:4 doi:10.1186/1759-2208-3-4.



## Acoustic sensor arrays for understanding bird communication. Identifying Cassin's Vireos using SVMs and HMMs

Julio G. Arriaga<sup>1</sup>, George Kossan<sup>2</sup>, Martin L. Cody<sup>2</sup>, Edgar E. Vallejo<sup>1</sup> and Charles E. Taylor<sup>2</sup>

<sup>1</sup>ITESM Campus Estado de México, Atizapán de Zaragoza, 52926, México

<sup>2</sup>University of California, Los Angeles, Los Angeles, CA, 90095, USA

taylor@biology.ucla.edu

### Abstract

In this paper, we present a series of experiments on the automated classification of Cassin's Vireo individuals from song phrases using support vector machines and from sequences of song phrases using hidden Markov models. Experimental results show that accurate classification of bird individuals can be achieved using these two different levels of description of bird songs.

### Introduction

Understanding the structure and function of bird songs is a long-sought goal in ecology research. Recent advances in sensor arrays, machine learning and computational linguistics finally make the achievement of this goal feasible. Understanding bird songs may also prove helpful in guiding the construction of artifacts that possess high-level communication abilities.

Over the last few years we have collected very large amounts of bird song recordings from acoustic sensor arrays in a variety of natural settings. This data have been processed by localizing source with beamforming, then filtering out noise, identifying events of interest, and then classifying them according to species and individual, and combining that with behavioral observations in a large database.

Our previous work on acoustic classification of birds has been successful at recognizing several species of antbirds and antbird individuals in a Mexican rainforest, Vallejo and Taylor (2009), Trifa et al. (2008). These birds possess quite simple, but distinctive, songs which are thought to be innate. In contrast, songbirds have a vocal organ highly developed that normally produce the relatively long and complicated vocalizations which are usually learnt, Catchpole and Slater (2008).

Particularly, the work presented here aims at exploring to what extent the methods we have used in the past are able to address and conduct the classification of songbird individuals. Further, it would be very useful for our research goals to understand the classification capabilities and limitations of sensor arrays when dealing with different levels of description of bird songs –song phrases, sequences of song

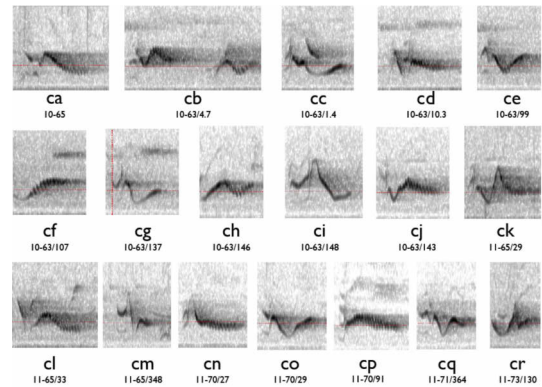


Figure 1: A subset of the CaVi phrases

phrases, etc. Toward this end, here we explore on classification of Cassin's Vireo (CaVi) individuals from song phrases using support vector machines (SVMs) and from sequences of song phrases using hidden Markov models (HMMs).

### Identifying CaVi individuals from song phrases using SVMs

The species of birds in our analysis have been the Cassin's Vireo (*Vireo cassinii*) a North American songbird, ranging from southern British Columbia in Canada through the western coastal states of the United States. The song consists of sequences of short, rough whistled phrases of several notes. The songs used in this work were recorded from April 2010 to July 2012, by Martin L. Cody. Examples of the CaVi songs are posted on <http://taylor0.biology.ucla.edu/al/bioacoustics/>.

A collection of 65 different phrases was identified by visual inspection of the sonograms. The sonograms of some of the CaVi phrases are in Figure 1, above. An example of extracted song grammar for a sample of the dataset is described by the Markov chain of figure 2. Samples of 12-53 phrases from each of the 12 individuals were included in the dataset. The sonogram of each phrase was measured for 124 traits using the Marsyas software package, Tzanetakis and Cook

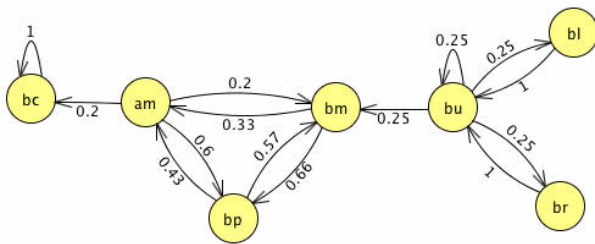


Figure 2: Markov chain of CaVi song. The states correspond to song phrases and the arrows indicate the transition probabilities among phrases

(1999) so that each song was represented by a vector. From these vectors, principal components were extracted and represented by a vector of 26 principal components.

Classification of individuals by SVMs was conducted using the Weka package, Witten et al. (2011). A radial basis function (RBF) kernel was used for the experiments. 10-fold cross-validation was conducted to find appropriate kernel parameters. Training was performed using the obtained kernel parameters on the training set. Testing was conducted using data samples not included in the training set. The classification results obtained in our experiments are in Figure 3.

### Identifying Cassin's Vireo individuals from sequences of song phrases using HMMs

Samples of 44 songs represented by sequences of phrases from each of the 12 individuals were included in the data set. Classification of individuals by HMMs was conducted using the Accord package <http://accord.googlecode.com/>. The Baum-Welch training algorithm was used for the experiments. A collection of training and validation experiments was conducted using this dataset. The classification results obtained in our experiments for each of the CaVi 12 individuals are presented in Figure 3. Results for HMMs are shown in gray, while those for SVMs in black. In nearly all cases the precision of classification was high - 90% or better. The HMMs appear to achieve slightly higher precision for practically all phrase types except for two, where there results were quite poor, only 40-50%, while the SVM results remained high throughout.

### Conclusions

Here we show that is feasible to discriminate songbirds individuals with similar accuracy from different levels of descriptions of birds songs –on the one hand, using their spectral and temporal acoustic features, and on the other hand, using the composition of their sequences of phrases.

The results presented are currently being analyzed by computational-linguistic tools to identify the syntax of the songs, and combined with information about the context in

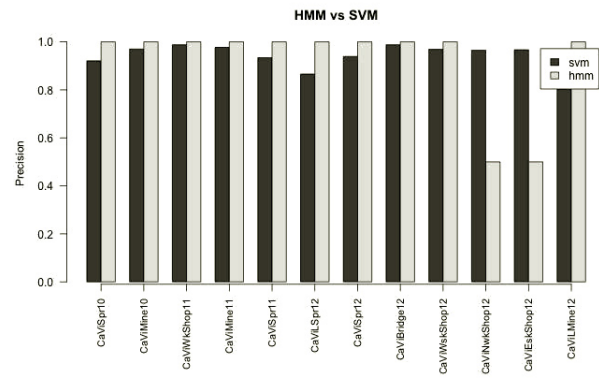


Figure 3: Classification of CaVi individuals. HMMs are shown in gray, those for SVMs in black

which they occurred, then analyzed by new software methods to identify the meaning of those songs. These methods will draw inferences from those meanings and explore consequences for individual and community ecology.

We believe this work will contribute to the recognition of very sophisticated signaling strategies and syntactic structures in non-human species. In addition, the work reported here can contribute to expand the range of engineering with voice recognition and classification, which so far has been restricted almost exclusively to humans. They hold promise to elucidate the fundamental properties of bird language. These results could then be useful to make progress on enabling high-level communication in artificial agents.

### Acknowledgements

This work was supported by the US National Science Foundation under Award Number 1125423 and by Consejo Nacional de Ciencia y Tecnología under Award Number I010/214/2012.

### References

- Catchpole, C. K. and Slater, P. J. B. (2008). *Bird Song. Biological Themes and Variations. Second Edition*. Cambridge University Press.
- Trifa, V. M., Kirschel, A. N. G., Taylor, C. E., and Vallejo, E. E. (2008). Automated species recognition of antbirds in a mexican rainforest using hidden markov models. *The Journal of Acoustical Society of America*, 123:2424–2431.
- Tzanetakis, G. and Cook, P. (1999). Marsyas: a framework for audio analysis. *Organized Sound*, 14:169–175.
- Vallejo, E. E. and Taylor, C. E. (2009). Adaptive sensor arrays for acoustic monitoring of bird behavior and diversity: preliminary results on source identification using support vector machines. *Artificial Life and Robotics*, 14:485–488.
- Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining. Third Edition*. Morgan Kaufmann.

## **Prebiotic Organic Microstructures as Model Protocells**

Henderson Cleaves and Neil Anthony

### **Abstract**

A variety of prebiotic syntheses starting from reduced and gas precursors, or small reactive organic intermediates produce a variety of micron and sub-micron sized organic microstructures, including spheres, filaments and toroids. Many of these structures are hollow, and they display dynamic and reversible self-assembly. We report here some of their physical characteristics that might be compatible with proto-cellular evolution.

## EMANN - a model of emotions in an artificial neural network

Ronald Thenius, Payam Zahadat and Thomas Schmickl

Artificial Life Laboratory, University of Graz, Austria.  
ronald.thenius@uni-graz.at

### Abstract

With the work at hand we want to present a model of a neural system, that is influenced by emotions. This model is based on the state of the art concept of artificial neural networks (ANN) which we improve by adding 'artificial emotions'. The way of the implementation of emotions is based on the research results regarding the biological and biochemical processes modulating neural cells in animals and humans. The described modulation takes place not only on the level of synapses, but also on the level of calculations happening on the membrane of the cell, or the node of the ANN, respectively. The suggested model also includes the biological fact, that neuro modulatory glands (e.g., hypophysis) are mainly controlled by the neuronal system itself. The resulting proposed system is named EMotional Artificial Neural Network (EMANN). It shows that EMANN has different abilities, compared to ANNs without emotions.

### Introduction

This work is mainly inspired by the work of Fellous (Fellous, 1999; Fellous et al., 2002; Fellous, 2009; Arbib and Fellous, 2004), who discussed the necessity of emotions in artificial systems. Fellous and his co-authors argue, that models of emotions can not only be used in artificial systems as a tool (e.g., to organise activity levels of different tasks with a single robot), but also serve as a model for emotions in animals and humans. Fellous described his ideas and requirements about the implementation of emotions in artificial systems as follows:

- Emotions should not be implemented as a separate, specialized module in charge of computing an emotional value on some dimensions.
- Emotions should not simply be the result of cognitive evaluations.
- Emotions are not linear (or non linear) combinations of some pre-specified basic emotions.
- Emotions should be allowed to have their own temporal dynamics, and should be allowed to interact with one another.

With this paper we propose an advanced concept of Artificial Neural Networks (ANN, described in detail in the following), using biological observations as guideline. Please note, that the proposed model was developed for the purpose to simulate emotions, and to investigate features of the interplay of biochemical, neuromodulatory processes with neural system. If and to what extent this is advantageous for technical solutions is topic of ongoing research (as mentioned below).

### Short overview over the state of the art of ANNs

The concept of ANNs was developed in the first half of 20th the century (McCulloch and Pitts, 1943; Rosenblatt, 1958). Since then several variations of ANNs were developed (Jain et al., 1996), be it differential equation based ANNs (CTRNN, by Beer (1995)) or models of spiking ANNs (Hindmarsh and Rose, 1984).

In all this mentioned methods, information is filtered by the ANN without any feedback of the information on the method of processing: The activity of one node influences the activity of another node via the connection of the two nodes. Via this signal transduction the signal is modulated by the weight associated with the regarding link. Manipulations of the process of filtering usually takes place on the level of synapses, and is done during the learning (or evolving) process. Only a few works exist, which try to modulate the process of data processing itself, based on input or output of the network (Neal and Timmis, 2003; Timmis et al., 2009). To our knowledge, no concept was suggested, in which the way of calculation is modulated and controlled by the network itself. From a biological point of view this ability of self-modulation is highly relevant in biological neural networks, and its implementation will add many new features to the concept of ANNs.

### Emotions and neurons in biological brains

On a biological level, moods and emotion can be understood as the condition an animal is in, based on an internal chemical (mostly hormonal) situation. This internal chemical situation modulates mostly all physiological and neural processes, what comes along with a modulation of as-



sociated behavioural phenomena. Examples for this are love (Carter, 1998; Young et al., 2001), fear (LeDoux, 2000; Derntl et al., 2009) and stress (Vermetten and Bremner, 2002). This way the moods influence as well “low-level” social behaviours (e.g., aggression (Pope et al., 2000)) as well as “high level” social behaviours (e.g. trust) (Kosfeld et al., 2005; Donaldson and Young, 2008; Guastella et al., 2008) etc. On a cellular/chemical level the emotional status is the result of a feedback-loop between the neural system and the hormonal system: On one hand the hormones influence the activity of the biological neural system (Derntl et al., 2009; Joëls, 1997), on the other hand the highest unit in the hierarchical cascade of interacting hormone glands (hypophysis, *glandula pituitaria*) is influenced by the central nervous system via the hypothalamus. Please note, that not all emotions emerge from such a feedback loop in the first place, but can also be triggered by other sources, e.g., by the genome directly.

Based on this phenomenon we developed the EMANN system (EMotional Artificial Neural Network), which improves the classical ANN paradigm (Jain et al., 1996) by the ability to develop hormone glands, that influence the abilities of the neural nodes. In this context the “mood” of an EMANN can be understood as the current hormonal status of the network. How much this “mood” influences the current behaviour of the controlled agent is coded into the individual responsiveness of the nodes of the EMANN.

## Method

### The concept

The suggested model is based on standard ANNs (Jain et al., 1996) which are improved by the features of a simulated neuromodulatory hormonal system. The simulated hormonal glands emit a number of virtual hormones, which represent all types of neuromodulators in a biological brain. These virtual hormones influence the behaviour of the individual nodes (which represent the individual cells within a biological neural system), including the synapses, the functions which sum up the inputs of the node, and the output function (which is analogous to the function of the axon hill of the neural cell). In return, the hormone glands are controlled by the neural network: each cell within the described network is able to emit hormones, which turns each node in the network into a possible hormone gland. To enable a virtual cell to react to a given hormonal level, hormonal receptors for every part of the virtual cell are simulated. These hormonal receptors are analogous to surface hormonal receptors in the membrane of biological neurons. A schematic drawing of the concept of ANN and EMANN is depicted in the Figures 1 and 2.

Throughout this paper, ‘node’ and ‘cell’, and also ‘edge’ and ‘synapse’ are used as synonyms. The main difference between the nodes of an ANN and an EMANN is that EMANN does not only map information from an input to an

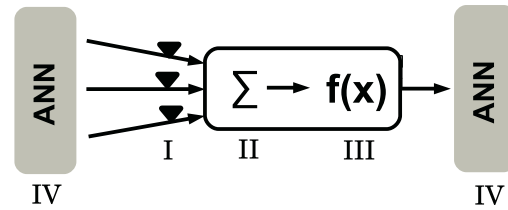


Figure 1: Schematic drawing of a single node in a state-of-the-art artificial neural network (ANN). The input of a node (*I - III*), which represents cells in a biological neural network, comes from other nodes of the ANN (*IV*) and/or from the input of the ANN. The weights of the edges between the nodes (*I*), representing synapses in the biological counterpart, are usually adapted to a given task by learning algorithms or by using artificial evolutionary methods. The net-function (*II*), modelling the processes in the dendritic part of a biological neuron, sums up the weighted inputs of the node. The out-function (*III*), from a functional point of view the analogon to the axon-hill in the biological neural cell, calculate the new status of the node. For more details about the state of the art in ANNs see (Jain et al., 1996). The solid arrows indicate the flow of information within the system, round edged triangles indicate the weights (synapses) of the node.

output level, but at the same time elements of a feedback system, that change the behaviour of the cells based on dynamic hormone levels. The “out-function” of the ANN differs from the EMANN analogon which is called “Hill-function” since the later is strongly influenced by hormones during runtime, but the former is usually static during runtime.

### From the biological concept to a mathematical representation

As represented in Figure 2, inputs to a cell of an EMANN system first pass through synapses where each synapse contains a weight that can be influenced by hormones (Figure 2:*I*). The value from the synapse then passes a net-function (represented by  $g(X)$  in Figure 2:*II*) and the total value of the net-outputs from all the input synapses is calculated. Finally, this sum passes a hill-function (represented by  $f(X)$  in Figure 2:*III*). The parameters of both net-functions and hill-functions are influenced by hormones.

In the following formal representations,  $W_{i,j}$  represents the weight of  $j$ th input synapse of cell  $i$ . Its value is specified as:

$$W_{i,j}(H) = \theta_{i,j} + \sum_h \phi_{i,j,h} H_h \quad (1)$$

where  $H$  is the set of hormone levels in the system and  $H_h$  is the level of hormone  $h$ .  $\theta_{i,j}$  is a constant weight for  $j$ th synapse of cell  $i$  and  $\phi_{i,j,h}$  is the responsiveness of the synapse to hormone  $h$ .

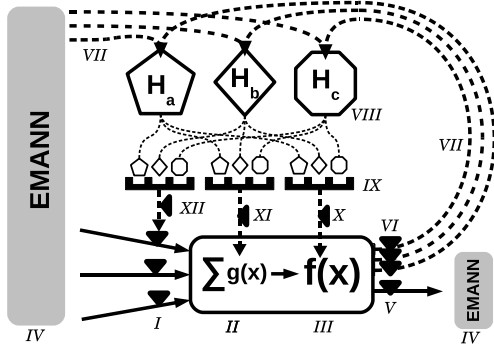


Figure 2: Schematic drawing of a single cell (analogon to the nodes in a state-of-the-art ANN) in an Emotional Artificial Neural Network (EMANN). In contrast to a state-of-the-art ANN (figure 1) the cell of the EMANN (I-III) is able not only to receive information from the EMANN (IV), or send information to it, but also to produce hormones (VIII). These hormones are represented by global variables, that sum up the hormone output of one time step of all cells of the EMANN (VII, equ. 12). Each cell is influenced by these hormones (IX), be it on the synaptic level (XII, see equ. 1) on the level of the net-function (XI, see equ. 2) or on the level of the hill function (X, see equ. 6). The degree, in which a cell functions as a neural cell or as a gland for a given hormone is determined by weights (V, VI, see equ. 13 and equ 11). Due to the interaction of cellular activity and hormone levels, a feedback arises, that allows to change the method of information processing, based on the information itself. Solid arrows indicate the flow of information within the EMANN (by which the input of the neural net is mapped to the neural output), the dotted lines indicate hormonal modulatory pathways.

An input to cell  $i$  from its  $j$ th synapse is represented by  $X_{i,j}$ . The input passes through the synapses and the net-function  $Net_{i,j}$  that is described by:

$$Net_{i,j}(X, H) = Netk_i(H) \times f(W_{i,j}(H)X_{i,j}) + Netd_i(H), \quad (2)$$

$$Netk_i(H) = \beta_i + \sum_h \eta_{i,h} H_h, \quad (3)$$

$$Netd_i(H) = \alpha_i + \sum_h \kappa_{i,h} H_h, \quad (4)$$

$$f(x) = \begin{cases} 1 & x < -1 \\ -1 & x > 1 \\ x & else \end{cases} \quad (5)$$

where  $\beta_i$  and  $\alpha_i$  are constant parameters of cell  $i$  and  $\eta_{i,h}$  and  $\kappa_{i,h}$  represent influences of hormone  $h$  on the net-function of the cell.

The net-outputs are then summed up and pass the hill-function and then produce output of the cell that is described as follows:

$$Hill_i(X, H) = Hillk_i(H) \times f\left(\sum_j Net_{i,j}(X, H)\right) + Hilld_i(H), \quad (6)$$

$$Hillk_i(H) = \lambda_i + \sum_h \sigma_{i,h} H_h, \quad (7)$$

$$Hilld_i(H) = \delta_i + \sum_h \rho_{i,h} H_h, \quad (8)$$

$$Y_i(X) = g(Hill_i(X, H)) \quad (9)$$

where  $g(x)$  is a unit step function.  $\lambda_i$  and  $\delta_i$  are constant parameters of cell  $i$  and  $\sigma_{i,h}$  and  $\rho_{i,h}$  represent influences of hormone  $h$  on the hill-function of the cell.

The output of a hill-function that is produced based on inputs to the cell can be summarized as:

$$Y_i = g\left(\left(\lambda_i + \sum_h \sigma_{i,h} H_h\right) \times f\left(\sum_j \left[\left(\beta_i + \sum_h \eta_{i,h} H_h\right) \times f\left(\left(\theta_{i,j} + \sum_h \phi_{i,j,k} H_h\right) X_{i,j}\right) + \left(\alpha_i + \sum_h \kappa_{i,h} H_h\right)\right] + \left(\delta_i + \sum_h \rho_{i,h} H_h\right)\right)\right) \quad (10)$$

The output of the cell then is calculated as a factor of the output of the hill-function as:

$$output_i = neurality_i \times Y_i \quad (11)$$

The overall amount of hormone  $h$  in the system is calculated as follows:

$$H_h = \sum_i H_{i,h}, \quad (12)$$

$$H_{i,h} = glandity_{i,h} \times Y_i \quad (13)$$

where  $H_{i,h}$  is the share of cell  $i$  in production of hormone  $h$  and  $glandity_{i,h}$  is a parameter representing the production factor of hormone  $h$  in the cell.

By deactivating the parameters of different modules of this system, e.g. net-function, we get simplified versions of the system.

### The task

To test, if the proposed model has features, that differ from a “non-emotional” neural network, we developed a task, to test the very basic features of the system.

The implemented task is to solve the following equation:

$$output(A, B) = \begin{cases} A & \text{if } A + B > 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

A simplified version of EMANN which uses the hill-function is implemented. A population of randomly generated ANNs and a population of randomly generated EMANNs are evolved for the task. The experiment is repeated for 100 independent runs in each case. The structures of the implemented ANN and EMANN are represented in Figure 3. The genomes are a sequence of parameters for each network and a standard genetic algorithm is used as the evolutionary algorithm. Experimental settings are summarized in Table 1.

Table 1: Experimental settings

|                       |                      |
|-----------------------|----------------------|
| number of cells       | 4                    |
| number of synapses    | 4                    |
| number of hormones    | 1                    |
| active-modules        | <i>hill-function</i> |
| population size       | 250                  |
| number of generations | 25000                |
| mutation rate         | 0.7                  |
| mutation probability  | 0.2                  |

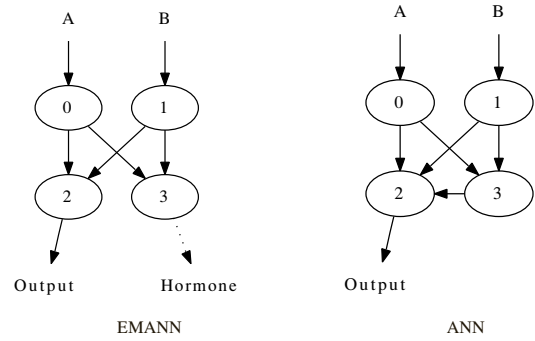


Figure 3: Network structures for the given task. The ANN, as well as the EMANN, have two inputs, and one neural output (as described in the section). Besides this, the EMANN has a hormonal output, that influences the *hill-function* (equ. 7) of all cells in the network.

### Results

The test of the EMANN concept in the task described above showed that the EMANN is able to generate higher fitness values in an evolutionary run comparing an ANN. The experiment was performed several times. An exemplary evolutionary run is depicted in figure 4. It shows, that both ANN and EMANN have a very fast increase of the fitness level reached within the first few generations. ANN reaches its maximum fitness of about 0.6 with the first few generations, while EMANN increases its reached fitness throughout the whole experiment.

Repeating the experiment several times showed that the better results from EMANN paradigm in comparison to the used state-of-the-art ANN (see figure 5 for more details) is statistically significant.

### Discussion

In his work Fellous (Fellous, 2004) argues that the implementation of emotions in an artificial system may not only be useful as a model for emotional processes in real-world lifeforms, but also may have advantages for purely artificial systems. As depicted in figure 5 we can show these assumptions are valid for the test described in using an EMANN and comparing it to a state-of-the-art ANN.

Regarding the biological counterparts, it is interesting from a biological point of view that state of the art ANN paradigms in most instances omit the fact, that in biological neural control systems are massively modulated, not to say ‘controlled’ by emotions. This phenomenon can be found in both, ‘regular’ emotions (as already mentioned above) or emotions based on diseases (e.g., depression, as described by Harding et al. (2004)). These modulations of the nervous system and the associated behaviour can vary from a biasing of the behavioural pattern, to a complete change of be-

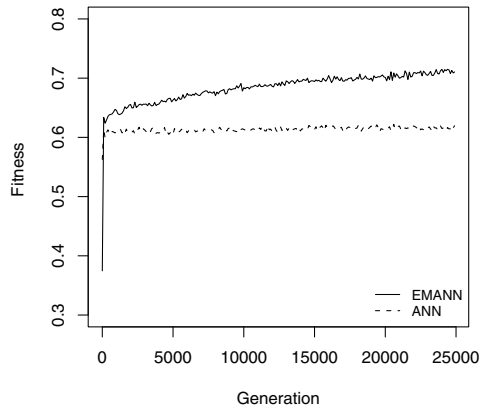


Figure 4: Exemplary evolutionary run of EMANN and ANN. It shows that the EMANN is able to solve the task better than the ANN. The task to solve is described in section . Curves indicate the maximum fitness value of every generation. Fitness is calculated according to equ. 14

haviour. Due to this, emotions can be understood as a kind of biological task-selection mechanism, from an engineering point of view. In literature the concept of emotions has been tested before, but always with a strict separation of hormonal system (e.g. hormonal glands) and the neural system: Gadanho and Hallam (2002) proposed a hormonal-neural system, which is described as a complex multilevel model, consisting of different compartments for cellular structures, neuromodulators and emotional activities. Neal and Timmis (2003) and Timmis et al. (2009) showed a system, in which the hormones interact with the neural system on synaptic level, but the hormone glands are a unit on its own on the input level of the system, with no ability to be influenced directly from the ANN. In the proposed EMANN paradigm, all actions and interactions, be it neurological, hormonal, hormone-gland related, neuromodulatory or emotional, take place on the level of cells and cellular interactions. Especially the ability of a cell to communicate with other cells on a local level (via neurotransmitters/synapses) or on a global level (via hormones) allows more complex activity patterns than a state of the art ANN. Further the cells in an EMANN have the ability not only to exchange information between cells (equ. 9) but also to send commands regarding changes in the information processing within the network (equ. 13) to groups of other cells within the EMANN. This results in a control system, which has the ability to modulate parts of the information processing system by itself, depending on the input of the EMANN.

Regarding the suggestions for the design of robot-emotions of Fellous, described in (Fellous, 2004), are re-

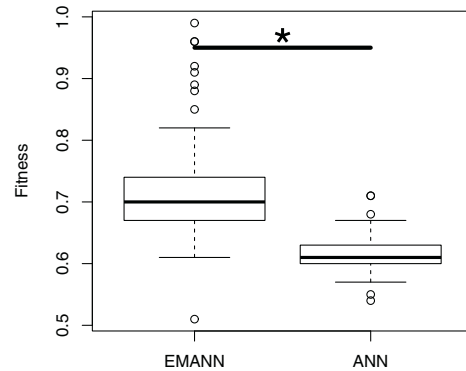


Figure 5: Maximum fitness values of last generation of 100 independent runs with 25000 generations. It shows that the EMANN performs significantly better in the given task compared to an ANN (\*: $p < 0.05$ ; Wilcoxon signed-rank test, unpaired data, "two.sided"-hypothesis, program used for statistical analysis: "R"). Box-plots indicate median and quartiles, whiskers indicate minimum and maximum, circles indicate outliers. Fitness is calculated according to equ. 14

flected in our work:

- *Emotions should not be implemented as a separate, specialized module in charge of computing an emotional value on some dimension.* In the EMANN paradigm hormone glands and neural cells are interlinked on a cellular level. As displayed in figure 2, a cell can act as a neural cell (equ. 9) as well as a hormone gland (equ. 13).
- *Emotions should not simply be the result of cognitive evaluations.* In an EMANN the emotions are the result of the interplay of neural activity and the interaction of hormones (by the modulation of neural cells and glands). The actual level of hormones are the outcome of a complex self-organising feedback system.
- *Emotions are not linear (or non linear) combinations of some pre-specified basic emotions.* The hormones in the EMANN paradigm do not represent any predefined emotions. Hormones act as neuromodulators, which change the functionality of the controller (in this case a neural network). This modulation can be understood as emotion.
- *Emotions should be allowed to have their own temporal dynamics, and should be allowed to interact with one another.* In an EMANN the emotions are an effect of the hormone levels, which are a result of the collective gland activity of all EMANN cells. This way the temporal dynamics of the emotion can be at least of the same dimension



of the neural cells, or much lower, depending from the degree and kind of neural linking of the hormone gland cells. Please note, that in the EMANN paradigm, the temporal dynamic of the hormones can not be higher than the temporal dynamics of the neural cells, but it can be equal or lower. The interaction of hormones takes place via the cells, which are receptors of hormones as well as emitters of hormones.

The results, depicted above show, that the EMANN for the given task is able to outperform an ANN significantly (figure 5). The task selected for this test was, from the authors point of view, the most simple possible, to show the abilities of the EMANN in a well defined small scale experiment.

## Conclusion and Outlook

Our work presents the first results of investigations in a model of emotions in an neural network, the Emotional Artificial Neural Network (EMANN). The EMANN is a development based on state of the art artificial neural networks (ANN, for an overview see (Jain et al., 1996)), improved by the ability to get modulated in its functionality by emotions. The emotions are implemented according to biological examples, inspired by the ideas presented in (Fellous, 2004). It showed, that an EMANN is able to outperform a state-of-the-art ANN in the given test. This shows, that the simulated emotions can have a positive effect on the performance of the ANN.

This paper is the first one in a row of investigations about the features of emotions in artificial neural networks. In the next step we will investigate, how learning algorithms based on emotions (inspired from the biological counterparts) can be used with an EMANN controller. We plan to investigate, how this system of neural, and hormonal feedbacks interacts under conditions of a changing environment, and how given evolutionary constraints (e.g., changing environments on an evolutionary time-scale) change the behaviour of the evolved EMANN controlled agents in complex environments. We further want to investigate, if the effect of genetically fixation of behaviour ("Baldwin effect" (Baldwin, 1896)) is influenced by the presence of emotions in the control system.

To investigate the evolutionary development of neural tissues and brain structures in an artificial system we plan to combine the EMANN paradigm with algorithms, that are able to shape morphological structures and controller structures in an evolutionary manner (Thenius et al., 2009b; Schmickl et al., 2011b; Kernbach et al., 2009; Thenius et al., 2010; Dauschan et al., 2011; Thenius et al., 2009a). To what extend the interplay of abiotic environment, social environment and behaviour can change the shape of brains in an artificial system (as described for biological systems, e.g., by Breedlove and Arnold (1980)) will be investigated in the

next step. We also plan to investigate the influence of emotions on the self-healing abilities (Thenius et al., 2011) of a controlling tissue in an artificial structured controller of an EMANN system.

One goal of the research in EMANN will be to investigate how the concept of emotions can be used in single autonomous robots or swarms of robots in complex environment (e.g., for (Schmickl et al., 2011a; Kernbach et al., 2008)). We plan, on a robotic swarm level, to share artificial hormones between spatially close agents (e.g., via short-range communication), what can result in a spatial organised task allocation or even spatially organised calculation processes based on interacting hormone levels inside a (big) swarm. The conditions, under which hormones are emitted within the swarm, and how the controllers of the swarm react to the given hormone level, can be as well engineered by hand, as well as shaped by an artificial evolutionary process.

Another goal is to investigate the role of emotions in the development of complex behaviours in an eusocial system, e.g., behaviours comparable to the BEECLUST algorithm (Schmickl et al., 2008; Bodi et al., 2012; Hereford, 2011).

To what extend the proposed system can be used as a model for biological and psychological processes (as suggested by Fellous in (Fellous, 2004)) will also be investigated in the future.

## Acknowledgements

This work was supported by: EU-IST-FET project SYMBRION, no. 216342; EU-ICT project REPLICATOR, no. 216240; EU-ICT 'CoCoRo', no. 270382; EU-ICT 'ASSIS-Ibf', no. 601074; Austrian Science Fund (FWF) research grants: P19478-B16 and P23943-N13 (REBODIMENT)

## References

- Arbib, M. A. and Fellous, J.-M. (2004). Emotions: from brain to robot. *Trends in Cognitive Sciences*, 8(12):554–561.
- Baldwin, M. J. (1896). A new factor in evolution. *The American Naturalist*, 30(354):441–451.
- Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3:469–509.
- Bodi, M., Thenius, R., Szopek, M., Schmickl, T., and Crailsheim, K. (2012). Interaction of robot swarms using the honeybee-inspired control algorithm beeclust. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):87–100.
- Breedlove, S. M. and Arnold, A. P. (1980). Hormone accumulation in a sexually dimorphic motor nucleus of the rat spinal cord. *Science*, 210(4469):564–566.
- Carter, C. S. (1998). Neuroendocrine perspectives on social attachment and love. *Psychoneuroendocrinology*, 23(8):779–818.
- Dauschan, M., Thenius, R., Schmickl, T., and Crailsheim, K. (2011). Using virtual embryogenesis multi-robot organisms. In *International Conference on Adaptive and Intelligent Systems, ICAIS'11, Klagenfurt, AT, September 06-08, 2011. Proceedings*, pages 238–247.

- Derntl, B., Windischberger, C., Robinson, S., Kryspin-Exner, I., Gur, R. C., Moser, E., and Habel, U. (2009). Amygdala activity to fear and anger in healthy young males is associated with testosterone. *Psychoneuroendocrinology*, 34(5):687–693.
- Donaldson, Z. R. and Young, L. J. (2008). Oxytocin, vasopressin, and the neurogenetics of sociality. *Science*, 322:900–904.
- Fellous, J.-M. (1999). The neuromodulatory basis of emotion. *The Neuroscientist*, 5(5):283–294.
- Fellous, J.-M. (2004). From human emotions to robot emotions. In Hudlicka, E. and Canamero, L., editors, *Architectures for Modeling Emotion: Cross-Disciplinary Foundations, AAAI Spring 2004 symposium, Stanford*. American Association for Artificial Intelligence.
- Fellous, J.-M. (2009). Emotion: Computational modeling. In Squire, L. R., editor, *Encyclopedia of Neuroscience*, volume 3, pages 909–913. Oxford: Academic Press.
- Fellous, J.-M., Armony, J. L., and LeDoux, J. E. (2002). Emotional circuits and computational neuroscience. In Arbib, M. A., editor, *The Handbook of Brain Theory and Neural Networks*. The MIT Press, 2 edition.
- Gadanh, S. C. and Hallam, J. (2002). Robot learning driven by emotions. *Adaptive Behavior*, 9(1):42–64.
- Guastella, A. J., Mitchell, P. B., and Dadds, M. R. (2008). Oxytocin increases gaze to the eye region of human faces. *Biological psychiatry*, 63(1):3–5.
- Harding, E. J., Paul, E. S., and Mendl, M. (2004). Animal behaviour: Cognitive bias and affective state. *Nature*, 427:312.
- Hereford, J. M. (2011). Analysis of BEECLUST swarm algorithm. In *Proc. of the IEEE Symposium on Swarm Intelligence (SIS 2011)*, pages 192–198. IEEE.
- Hindmarsh, J. L. and Rose, R. M. (1984). A model of neuronal bursting using three coupled first order differential equations. *Proceedings of the Royal society of London. Series B. Biological sciences*, 221(1222):87–102.
- Jain, A. K., Mao, J., and Mohiuddin, K. (1996). Artificial neural networks: A tutorial. *IEEE Computer*, 29:31–44.
- Joëls, M. (1997). Steroid hormones and excitability in the mammalian brain. *Frontiers in neuroendocrinology*, 18(1):2–48.
- Kernbach, S., Hamann, H., Stradner, J., Thenius, R., Schmickl, T., Crailsheim, K., van Rossum, A., Sebag, M., Bredeche, N., Yao, Y., Baele, G., de Peer, Y. V., Timmis, J., Mohktar, M., Tyrrell, A., Eiben, A., McKibbin, S., Liu, W., and Winfield, A. F. (2009). On adaptive self-organization in artificial robot organisms. In *The First International Conference on Adaptive and Self-adaptive Systems and Applications (ADAPTIVE'09)*. IEEE Press.
- Kernbach, S., Meister, E., Schlachter, F., Jebens, K., Szymanski, M., Liedke, J., Laneri, D., Winkler, L., Schmickl, T., Thenius, R., Corradi, P., and Ricotti, L. (2008). Symbiotic Robot Organisms: REPLICATOR and SYMBRION Projects. In Madhavan, R. and Messina, E., editors, *Proceedings of the Performance Metrics for Intelligent Systems Workshop (PerMIS)*, pages 62–69. NIST Special Publication 1090.
- Kosfeld, M., Heinrichs, M., Zak, P. J., Fischbacher, U., and Fehr, E. (2005). Oxytocin increases trust in humans. *Nature*, 435:673–676.
- LeDoux, J. E. (2000). Emotion circuits in the brain. *Annual review of neuroscience*, 23:155–184.
- McCulloch, W. S. and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(4):115–133.
- Neal, M. and Timmis, J. (2003). Timidity: A useful emotional mechanism for robot control? *Informatica*, 27(4):197–204.
- Pope, H. G., Kouri, E. M., and Hudson, J. I. (2000). Effects of supraphysiologic doses of testosterone on mood and aggression in normal men: A randomized controlled trial. *Archives of General Psychiatry*, 57(2):133–140.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.
- Schmickl, T., Thenius, R., Christoph, M., Timmis, J., Tyrrell, A., Read, M., Hilder, J., Halloy, J., Campo, A., Stefanini, C., Manfredi, L., Orofino, S., Kernbach, S., Dipper, T., and Sutantyo, D. (2011a). Cocoro – the self-aware underwater swarm. In *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2011 Fifth IEEE Conference on*, pages 120–126.
- Schmickl, T., Thenius, R., Möslinger, C., Radspieler, G., Kernbach, S., and Crailsheim, K. (2008). Get in touch: Cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18(1):133–155.
- Schmickl, T., Thenius, R., Stradner, J., Hamann, H., and Crailsheim, K. (2011b). Robotic organisms: Artificial homeostatic hormone system and virtual embryogenesis as examples for adaptive reaction-diffusion controllers. In Kernbach, S. and Fitch, R., editors, *IROS 2011 Workshop-Reconfigurable Modular Robotics: Challenges of Mechatronic and Bio-Chemo-Hybrid Systems*.
- Thenius, R., Bodi, M., Schmickl, T., and Crailsheim, K. (2009a). Growth of structured artificial neural networks by virtual embryogenesis. In Kampis, G., Karsai, I., and Szathmáry, E., editors, *ECAL (2)*, volume 5778 of *Lecture Notes in Computer Science*, pages 118–125. Springer.
- Thenius, R., Bodi, M., Schmickl, T., and Crailsheim, K. (2010). Evolving artificial neural networks and artificial embryology. In Levi, P. and Kernbach, S., editors, *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*, pages 266–268. Springer-Verlag.
- Thenius, R., Dauschan, M., Schmickl, T., and Crailsheim, K. (2011). Regenerative abilities in modular robots using virtual embryogenesis. In *International Conference on Adaptive and Intelligent Systems, ICAIS'11, Klagenfurt, AT, September 06-08, 2011. Proceedings*, pages 227–237.
- Thenius, R., Schmickl, T., and Crailsheim, K. (2009b). Novel concept of modelling embryology for structuring an artificial neural network. In Troch, I. and Breitenacker, F., editors, *Proceedings of the MATHMOD*, pages 1821–1831.

- Timmis, J., Neal, M., and Thorniley, J. (2009). An adaptive neuro-endocrine system for robotic systems. In *IEEE Workshop on Robotic Intelligence in Informationally Structured Space. Part of IEEE Workshops on Computational Intelligence, Nashville*.
- Vermetten, E. and Bremner, J. D. (2002). Circuits and systems in stress. i. preclinical studies. *Depression and anxiety*, 15(3):126–147.
- Young, L. J., Lim, M. M., Gingrich, B., and Insel, T. R. (2001). Cellular mechanisms of social attachment. *Hormones and Behavior*, 40:133–138.

## Image Similarity Search using a Negative Selection Algorithm

Stein Keijzers<sup>1</sup>, Peter Maandag<sup>1</sup>, Elena Marchiori<sup>1</sup>, Ida Sprinkhuizen-Kuyper<sup>2</sup>

<sup>1</sup>Department of Computer Science, Faculty of Sciences, Radboud University Nijmegen, The Netherlands

<sup>2</sup>Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour, Centre for Cognition, The Netherlands  
skeijzers@home.nl, pmaandag@upcmail.nl, elenam@cs.ru.nl, i.kuyper@donders.ru.nl

### Abstract

The Negative Selection Algorithm is an immune inspired algorithm that can be used for different purposes such as fault detection, data integrity protection and virus detection. In this paper we show how the Negative Selection Algorithm can be adapted to tackle the similar image search problem: given a target image, images from a large database similar to the query have to be detected. Results of our experimental analysis indicate that the proposed algorithm is capable of detecting images similar to a target (self) image, given the right detectors. Source code and data used in the experiments are available on request.

### Introduction

The increasing storage capacity of modern disk devices allows to collect and distribute large-scale image data efficiently. As a consequence, an enormous amount of images is generated and made publicly available. This phenomenon has boosted research on search for similar images. Search by image as implemented for example by Google (2013) allows one to discover all sorts of content that is related to a specific image. Search results include similar images, webpages and even sites that include that picture. The main challenge is to develop efficient methods for achieving high image retrieval quality. Many techniques have been proposed for this task (see for instance Smeulders et al. (2000)).

A crucial step in image similarity search is the comparison of two images, typically a target image and an image from the database. Image distance measures are used for this task. An image distance measure compares the similarity of two images in various dimensions such as color, texture, shape, and others.

Our goal is to investigate whether a negative selection algorithm with simple techniques for comparing images is suitable for tackling the Image Similarity Search problem.

To this aim, we designate the target image as self data and then create detectors for anything that is not self. We consider a simple setting where color is used as the only feature to define distances between images. Color does not depend on image size or orientation, hence is easy to handle.

In order to analyze whether NSA is applicable to Image Similarity Search, two sets of experiments are conducted. First, we consider a publicly available dataset of holiday images and assess manually how good the results of NSA are. Second, we construct a specific dataset consisting of three classes and manually associate a class to each image. We perform a leave-one-out cross validation on the dataset. Specifically for each image, we consider it as target image and use the rest of the data to search for similar images. Then average precision of each target is computed and the mean results are analyzed.

Results of experiments indicate the suitability of negative selection for this task, considering the fact that we use only color as feature. We would like to stress that the goal of this paper is not to try to compete with algorithms such as those used by Google, but to show that NSA can be applied to search by image.

### Background

The original negative selection algorithm is inspired by the way that natural immune systems distinguish self from other (Forrest et al., 1994). When the body encounters a virus or any cells that do not belong to the body's own cells, white blood cells are sent out to react to this and to destroy the foreign cells. A specific type of these blood cells are the so-called T-Cells. One of the things these might do is destroy cells infected by viruses. However, one interesting property of these cells is that they somehow know how to discriminate between foreign tissue and your own body. The process that leaves only these T-cells alive is called negative selection (Dasgupta et al., 2011).

The T-cells are formed in the thymus. The thymus is a small organ that is located in front of the heart and near the sternum. In this organ there are a lot of proteins that belong to the body itself. When a T-cell is just formed it might attack these proteins. These immature T-cells that strongly bind to these self-antigens undergo a controlled programmed cell death, referred to as apoptosis (Stibor et al., 2005). We will not go into the details of this mechanism. What is important to observe is that generally only T-cells that attack



foreign antigens leave the Thymus. Thus, the T-cells that survive this process should be nonreactive to self-antigens, but attack antigens they don't know instead.

The Negative Selection Algorithm (NSA) (Forrest et al., 1994) is inspired by a main mechanism in the thymus that produces a set of mature T-cells capable of binding only to non-self antigens. Many variants of this mechanism exist, often exploiting other natural elements, see for instance Gao et al. (2008) and Shapiro et al. (2005). NSA has many applications, most notably in the field of fault and intrusion detection (Dasgupta and Forrest, 1995; Kim and Bentley, 2001; Taylor and Corne, 2003; Dasgupta et al., 2004). Another typical application is anomaly detection (Dasgupta and Majumdar, 2002; González and Dasgupta, 2003).

NSA basically consists of two steps, as shown schematically in Figure 1 and 2 for the scenario where the integrity of a data file or string has to be protected (Forrest et al., 1994). Using NSA, the first step then is to generate a set of detectors. Each detector is a string that does not match a predetermined substring of the protected data. For matching, usually a partial matching rule is defined, because it can be extremely rare that random strings that are generated exactly match the source data, even if these strings are small.

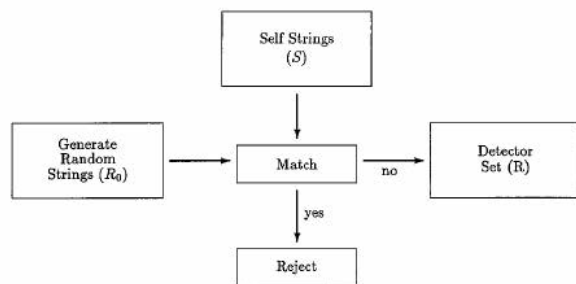


Figure 1: Detector Generation, taken from Forrest et al. (1994)

The second step is to continually monitor the data by comparing them with the detectors. If a detector is ever activated, a change is known to have occurred.

Although this approach might seem too simple to work, it is rather effective: a fairly small set of detector strings has a very high probability of noticing a random change to the original data (Forrest et al., 1994).

### NSA for image similarity search

We want to apply the NSA to image similarity search. To this aim, we designate the target image as self data and then create detectors for anything that is not self. For this to work we try to match pixels or pixel groups on each other and use the fraction of matching groups as a similarity measure. A match can be determined via a direct pixel to pixel comparison, but also via a similarity measure, like the one that will

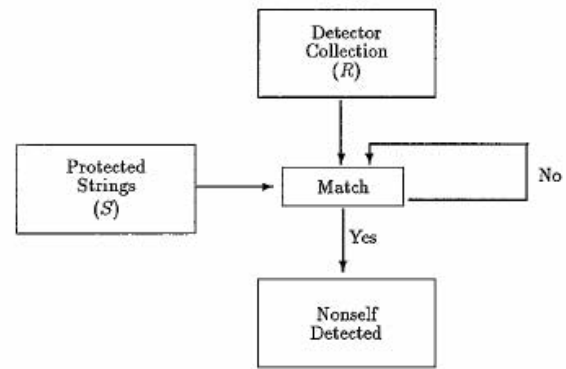


Figure 2: Monitor protected data, taken from Forrest et al. (1994)

be explained later.

Once we have generated detectors implementing these methods we can match them on the data set and retrieve dissimilarity values, which can then be inverted and used to identify images that are very similar to the target image.

In effect, we're trying an alternative approach to achieve what Google does with reverse image search. You can upload a picture to Google and Google returns a list of similar images on the web (Google, 2013). In our case we adapt the negative selection algorithm to accomplish this.

### Framework

The NSA algorithm is implemented using C++, because we needed to load and clear many Gigabytes of images in the RAM and this can be done efficiently in C++. It's also object-oriented, so producing a framework was easier.

To apply the algorithm we wrote a framework that handles file input and output. It includes a Detector super-class, which has initialization and detection functions by default. Any detector we implemented was able to provide implementations and extensions to this detector class, while maintaining the basic functionality needed for the algorithm.

Figure 3 illustrates the proposed method. The NSA algorithm (in the center) receives images (the target image and the database of images for searching) from an image-reader class. It then applies its detectors to those images to get a similarity measure.

### Detectors

The algorithm is strongly dependent on the detectors that are created. Each of the detectors implements a basic measure of similarity between two images. The idea here is that, with enough of these detectors, we can get a reasonable approximation for the actual similarity by combining the advantages of each detector and by canceling out the downsides of each single similarity measure. The advantage is then that running these simple detectors, possibly in parallel, would

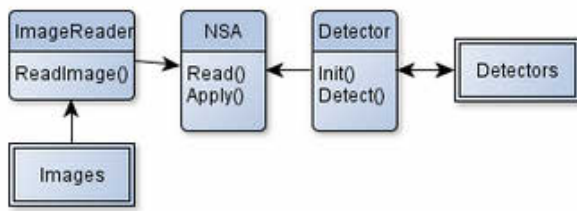


Figure 3: NSA framework

be much cheaper than running more complicated measures, such as those based on compact data structures and Earth Movers Distance (see e.g. Lv et al. (2004)).

**Direct pixel similarity (DPS)** One of the first ideas that comes to mind when implementing a detector is the direct pixel similarity (DPS). In this case we are working at the pixel level. So the most simple and obvious similarity measure is a direct RGB comparison of each pixel. If the pixels are the same they match, if they are different they do not match. All the matches are counted and normalized to one.

Obviously this is not a very good detector since it is rare that many pixels in a picture match exactly with the target picture. There might be subtle changes in brightness or other minor differences that are barely visible and that will negatively influence such similarity measures.

Therefore we weaken the similarity match by considering a range for each pixel value  $v$ , such that we still agree on a match if the value  $v$  of each color component is in the range  $[v - r, v + r]$ , for a given parameter  $r$ .

Furthermore, we define a match on a group of  $n$  by  $n$  pixels if for a certain threshold  $t$ ,  $t$  pixels in a group of the target image match the foreign image. In the end we count all matches and divide by the total amount of pixel groups that were compared to get the final similarity measure. In Figure 4 we can see how the proposed matching pixel group comparison works with  $n = 3$ ,  $t = 5$ , and  $r = 0$ . The pixels in row 1 up to 3 and column 1 up to 3 are directly compared. Every pixel in this group has the same value so the match count is 9. This is bigger than the threshold  $t = 5$  so these pixel groups match.

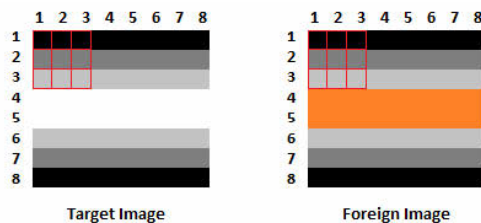
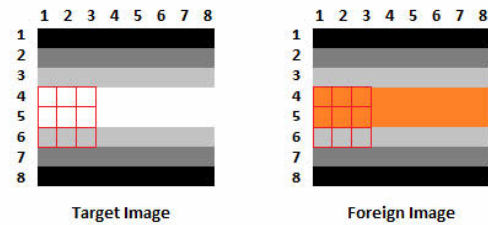
Figure 4: Matching pixel group comparison with  $n = 3$ ,  $t = 5$ 

Figure 5 shows an example of a pixel group that does not

match. The pixels in row 4 up to 6 and column 1 up to 3 are directly compared. But the pixels in row 4 and 5 have different values so there are only 3 pixels that match. Since  $3 < t$ , these pixel groups do not match.

Figure 5: Non-matching pixel group comparison with  $n = 3$ ,  $t = 5$ 

**Average color difference similarity (ACDS)** Another measure we consider incorporates the summed color difference of each color component. By dividing this summed difference by the absolute maximum difference possible, we get a similarity measure between 0 and 1: the average color difference similarity (ACDS).

The same matching rule as in the previous method can then be used on a group of pixels, but in this case we define the threshold  $t$  to be a value between 0 and 1. In this way we are basically saying that a pixel group of the foreign image has to match for  $100t$  percent with the target image.

This measure has as advantage that its a lot more flexible: if a picture is similar in part to the target image, but different in some others, it might still get a high score. A disadvantage is that, since comparing colors is not that easy, you can get some results, where colors are considered similar even though to our eyes they might not be. For instance, brown is very similar to grey from pure values, but to the human eye it would be different.

## Application

In our application we use the holiday data set (Jegou et al., 2008) that consist of 812 pictures that were made by people on their holidays, see Figure 6 for a snapshot. The collection is applicable to the problem at hand because it contains clusters of similar images, for example for outdoor environments, while maintaining a very broad category, in contrast to other datasets which are either too specific or too general (Toet et al., 2001; Deng et al., 2009; Chen et al., 2009). Indoor surroundings with similar characteristics can also be found, so we were able to identify similar images from this set. A downside of this dataset is that the original images vary in width and height so additional preprocessing on the data set had to be done to improve performance in both run-time and quality. We have resized all images to a smaller, equal size before the algorithm was run. Manual inspection of the results was used to assess their quality.



Figure 6: Snapshot of image dataset

We experimented with the length of detector bit strings as suggested in Forrest et al. (1994), namely powers of 2 ranging from 32 to 256 for this value, which in our case is equal to  $n^2$ . Forrest et al. (1994) used a partial matching function that measures an amount of contiguous matches. Since our implemented initial detectors only use direct pixel comparisons, we could not take over their values directly. Instead for DPS we have performed some experiments with values between 0.5 and 1 for the threshold  $t$ . A too high threshold causes bad results because it means that every pixel in a group has to be equal to the other group, while a too low threshold causes that every picture is more likely to be highly similar to another. A threshold  $t$  of  $0.75n^2$  in combination with  $n = 4$  yields good results for DPS.

For ACDS the threshold was also manually tuned. In this case a too high threshold results into every picture being similar, while a too low threshold returns a similarity of at most 0. A threshold of 0.1 in combination with  $n = 16$  seemed to be nicely in the middle for ACDS.

The optimal value for the color range  $r$  was also determined by experimenting. Small values quickly start to give back topmost results which have 0% similarity. If  $r$  is set too high then everything is being returned as similar. A value of  $r = 30$  gives good results for most pictures. An exception is the city target image (see below) for which a value of  $t = 15$  gives better results.

### Parameters

After conducting many experiments we settled for  $n = 4$ ,  $t = 0.75 * n^2$ ,  $r = 30$  for DPS and  $n = 16$ ,  $t = 0.1$  for ACDS.

In the following, we will show results for three target images in different types of environments: city, scenery and an anemone in sea.

### City

The city target image is shown in Figure 7. It is characterised by a high diversity in the image.



Figure 7: City Target Image



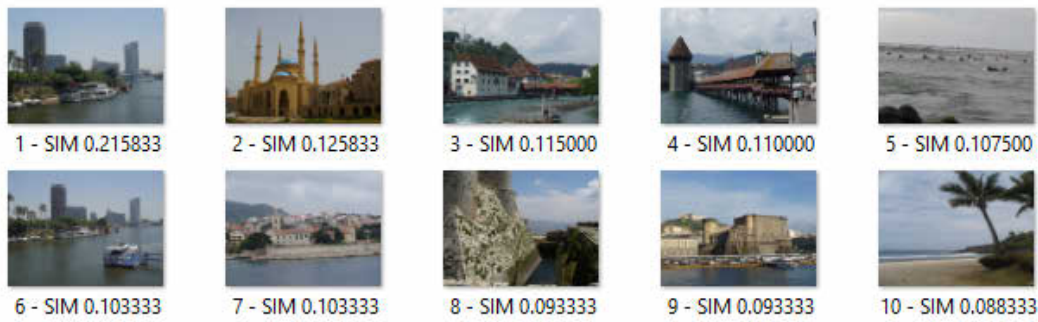
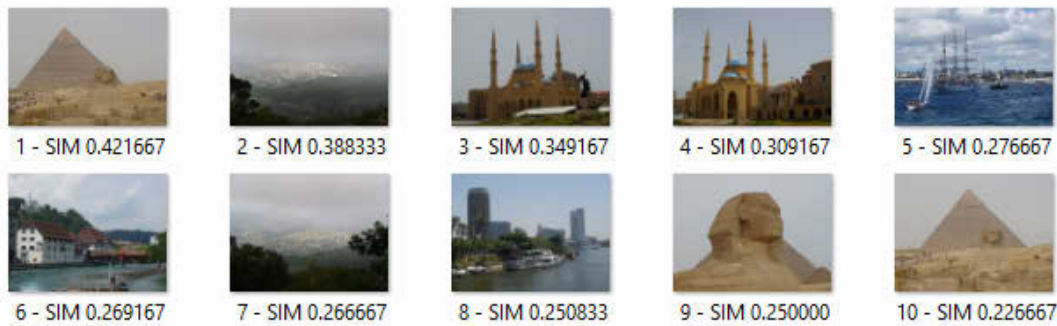
Figure 8: Best Results for City Direct Pixel Similarity with  $n = 16$ ,  $t = 0.75 * n^2$ ,  $r = 15$ Figure 9: Best Results for City Direct Pixel Similarity with  $n = 16$ ,  $t = 0.75 * n^2$ ,  $r = 30$ 

Figure 8 and Figure 9 show the results for the DPS detector with  $r = 15$  and  $r = 30$ , respectively. These results clearly demonstrate the effects of the range value. If the range is set too high then different colors are considered similar. In the case of  $r = 30$  the first result can be explained by the characteristics of the target image. There is the blue sky and a dark part of the city. The pyramid photo shows exactly this pattern. If the range is set to a narrower value  $r = 15$ , better results are obtained.

The results of the ACDS detector for the city image are similar to those in Figure 9. This detector clearly has problems with the diversity in the city view.

### Scenery

A scenery target image considered is shown in Figure 10. It is characterised by a bright sky in the upper half and a darker soil in the lower half.

Figure 11 shows the resulting images for Direct Pixel Similarity. We can see that the most similar images that are returned resemble the characteristics of a bright sky in the upper half and a darker substance in the lower part, which corresponds with core characteristics of the target image.

Figure 12 shows the resulting images for Average Color Difference Similarity. Here the resemblance between pictures is still pretty high, but less distinct. For example in the output images number 8 and 3 a less clear distinction



Figure 10: Scenery Target Image

between the upper half and the lower half can be observed.

### Anemone

An Anemone target image in a sea environment is shown in Figure 13. It is characterised by shades of green over the whole image without a clear separation between the sky and the ground as in the scenery setting.

Figure 14 shows the resulting images for Direct Pixel Similarity. The top three results are exactly what we want to see. The rest of the images also contain similar shades





Figure 11: Best results for Scenery Direct Pixel Similarity with  $n = 4$ ,  $t = 0.75 * n^2$ ,  $r = 30$

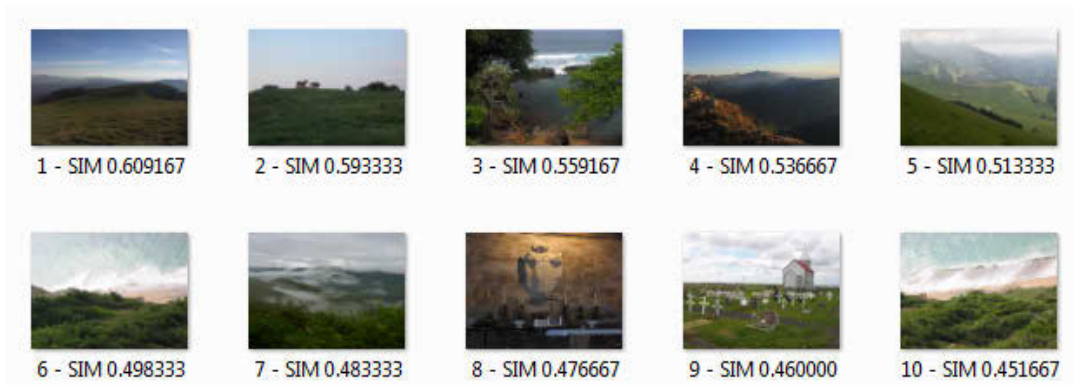


Figure 12: Best results for Scenery Average Color Difference Similarity with  $n = 16$ ,  $t = 0.1$



Figure 13: Anemone Target Image

of green as we would expect. However there are also less obvious hits such as 6, 8 and 10. This might be explained by the fact that the target picture has a lot of variation in pixel colors in each group.

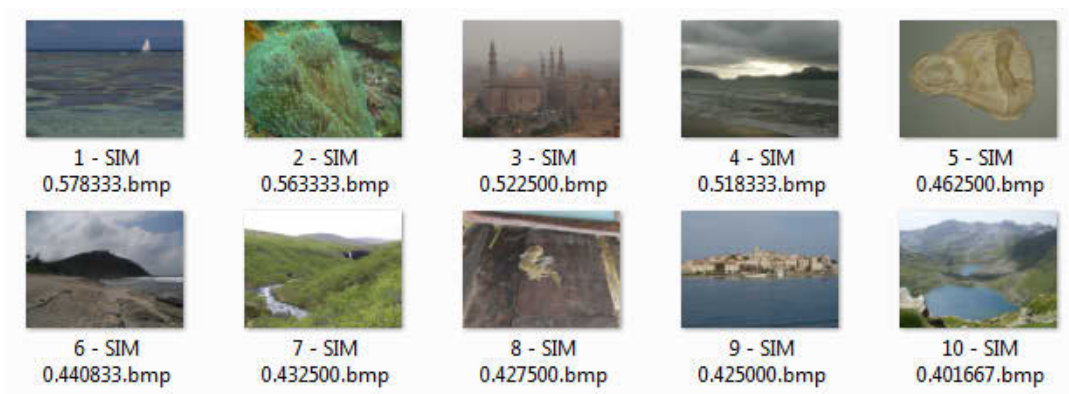
The Average Color Difference yields even worse results, see Figure 15, because it focusses on the average color in a group. This means that pictures which have many different

pixel colors within each group will have a higher probability of being similar to each other when they are averaged, which can yield strange results.

### Validation on labeled data

In order to assess the quality of the results automatically we consider a small dataset containing 60 images manually labeled according to three classes.

For each image, we remove it from the dataset and compute the average precision (see e.g. Müller et al. (2001)) over the entire ranking. Mean results over all images are reported in Table 1. In order to analyze the significance of these results, the average precision of each target image is compared to that of 1000 random rankings. Empirical p-values are then computed, as the fraction of times the average precision on random ranking was better than that on the ranking generated using NSA. The resulting p-values are 0.004 for ACDS and 0.001 for DPS. Results indicate that NSA is better than a (random) baseline method, that DPS and ACDS have similar variance, and that DPS performs better than ACDS.

Figure 14: Best results for Anemone Direct Pixel Similarity with  $n = 4, t = 0.75 * n^2, r = 30$ Figure 15: Best results for Anemone Average Color Difference Similarity with  $n = 16, t = 0.1$ 

| mean average precision<br>ACDS (std) | mean average precision<br>DPS (std) |
|--------------------------------------|-------------------------------------|
| 35.65% (3.39)                        | 41.82% (3.43)                       |

Table 1: Leave-one-out results on a manually labeled dataset with three classes. Mean average precision across the images; std denotes standard deviation.

## Discussion

This paper investigated the use of NSA for content-based image retrieval. We have introduced a simple method based on NSA and showed that it can achieve promising similarity search results on a collection of general-purpose images as assessed by human inspection.

In future work we plan to extend the proposed method by incorporating also detectors defined on dimensions such as texture and shape. This will allow to handle image collections of general-purpose images from various given categories. In this way, for each image, the sets of images from its category can be used as the ground truth in the evalu-

ation. Hence effectiveness measures such as average precision can be used to formally assess the performance of a method. This will allow us to perform a comparative assessment of the proposed method using state-of-the-art algorithms for this task.

The two detectors we have implemented are just two examples of what can be done to measure similarity. There are many more possibilities notably measures based on Earth Mover's Distance, which is a similarity measure between multidimensional distributions (Lv et al., 2004). Other similarity measures include contrast measurements, color matrices or even completely different approaches such as edge detection. If we can use these detectors together, possibly in parallel, the performance and resulting similarity might be improved even further.

Nevertheless the current results based on our simple method are already fairly similar to (manual) human eye selection, which is usually what a user of such a system would want. Our conclusion is therefore that Negative Selection Algorithms can indeed aid in creating a search system for image similarity search.

A number of aspects of this research remain to be investigated. For instance, adding non-determinism and random elements to detectors, thereby applying the algorithm more literally, could give more varied results, but also likely less precise. Detector coverage in such a non-deterministic scenario could then be improved by applying analytic methods (Ji and Dasgupta, 2005). Another possibility is increasing performance of this algorithm, either by better pre-processing of the data (see e.g. the methods described in the survey by Smeulders et al. (2000)), or by applying detectors in parallel.

## References

- Chen, M., Dhingra, K., Wu, W., Yang, L., Sukthankar, R., and Yang, J. (2009). PFID: Pittsburgh fast-food image dataset. In *Proceedings of the 16th IEEE international conference on Image processing, ICIP'09*, pages 289–292, Piscataway, NJ, USA. IEEE Press.
- Dasgupta, D. and Forrest, S. (1995). Tool breakage detection in milling operations using a negative-selection algorithm. Technical Report CS95-5, Department of Computer Science, University of New Mexico, Albuquerque, NM.
- Dasgupta, D., KrishnaKumar, K., Wong, D., and Berry, M. (2004). Negative selection algorithm for aircraft fault detection. In *Proceedings of the 3rd International Conference on Artificial Immune Systems, ICARIS*, volume 3239 of *Lecture Notes in Computer Science*, pages 1–13, Berlin Heidelberg. Springer.
- Dasgupta, D. and Majumdar, N. S. (2002). Anomaly detection in multidimensional data using negative selection algorithm. In *Proceedings of the 2002 Congress on Evolutionary Computation*, volume 2 of *CEC'02*, pages 1039–1044. IEEE.
- Dasgupta, D., Yu, S., and Nino, F. (2011). Recent advances in artificial immune systems: models and applications. *Applied Soft Computing*, 11(2):1574–1587.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2009, pages 248–255. IEEE.
- Forrest, S., Perelson, A. S., Allen, L., and Cherukuri, R. (1994). Self-nonspecific discrimination in a computer. In *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, pages 202–212. IEEE Computer Society Press.
- Gao, X.-Z., Ovaska, S. J., and Wang, X. (2008). A GA-based negative selection algorithm. *International Journal of Innovative Computing, Information and Control*, 4(4):971–979.
- González, F. A. and Dasgupta, D. (2003). Anomaly detection using real-valued negative selection. *Genetic Programming and Evolvable Machines*, 4(4):383–403.
- Google (2013). Inside search - google. <http://www.google.com/insidesearch/features/images/searchbyimage.html>.
- Jegou, H., Douze, M., and Schmid, C. (2008). Hamming embedding and weak geometry consistency for large scale image search. *Proceedings of the 10th European conference on Computer vision*. INRIA Holidays dataset available at <http://lear.inrialpes.fr/~jegou/data.php#holidays>.
- Ji, Z. and Dasgupta, D. (2005). Estimating the detector coverage in a negative selection algorithm. In *Proceedings of the 2005 conference on Genetic and evolutionary computation, GECCO '05*, pages 281–288, New York, NY, USA. ACM.
- Kim, J. and Bentley, P. J. (2001). An evaluation of negative selection in an artificial immune system for network intrusion detection. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2001*, pages 1330–1337, San Francisco, US. Morgan Kaufmann.
- Lv, Q., Charikar, M., and Li, K. (2004). Image similarity search with compact data structures. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management, CIKM '04*, pages 208–217, New York, NY, USA. ACM.
- Müller, H., Müller, W., Squire, D. M., Marchand-Maillet, S., and Pun, T. (2001). Performance evaluation in content-based image retrieval: overview and proposals. *Pattern Recogn. Lett.*, 22(5):593–601.
- Shapiro, J. M., Lamont, G. B., and Peterson, G. L. (2005). An evolutionary algorithm to generate hyper-ellipsoid detectors for negative selection. In *Proceedings of the 2005 conference on Genetic and evolutionary computation, GECCO '05*, pages 337–344, New York, NY, USA. ACM.
- Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380.
- Stibor, T., Mohr, P., Timmis, J., and Eckert, C. (2005). Is negative selection appropriate for anomaly detection? In *Proceedings of the 2005 conference on Genetic and evolutionary computation, GECCO '05*, pages 321–328, New York, NY, USA. ACM.
- Taylor, D. W. and Corne, D. W. (2003). An investigation of the negative selection algorithm for fault detection in refrigeration systems. In *Proceeding of Second International Conference on Artificial Immune Systems, ICARIS*, volume 2787 of *Lecture Notes Computer Science*, pages 34–45. Springer, Berlin Heidelberg.
- Toet, A., Bijl, P., and Valetton, J. M. (2001). Image dataset for testing search and detection models. *Optical Engineering*, 40(9):1760–1767.



# Immune-Inspired Error Detection for Multiple Faulty Robots in Swarm Robotics

HuiKeng Lau<sup>3</sup> and Iain Bate<sup>1,4</sup> and Jon Timmis<sup>1,2</sup>

<sup>1</sup>Department of Computer Science, University of York, UK,

<sup>2</sup>Department of Electronics, University of York, UK,

<sup>3</sup>Applied Computing Group, SKTM, UMS, Malaysia,

<sup>4</sup>Mälardalen Real-Time Research Centre, Mälardalen University, Västerås Sweden

hklau@ums.edu.my, {iain.bate, jtimmis}@york.ac.uk

## Abstract

Error detection and recovery are important issues in swarm robotics research, as they are a means by which fault tolerance can be achieved. Our previous work has looked at error detection for single failures in a swarm robotics scenario with the Receptor Density Algorithm. Three modes of failure to the wheels of individual robots was investigated and comparable performance to other statistical methods was achieved. In this paper, we investigate the potential of extending this approach to a robot swarm with multiple faulty robots. Two experiments have been conducted: a swarm of ten robots with 1 to 8 faulty robots, and a swarm of 10 to 20 robots with varying number of faulty robots. Results from the experiments showed that the proposed approach is able to detect errors in multiple faulty robots. The results also suggest the need to further investigate other aspects of the robot swarm that can potentially affect the performance of detection such as the communication range.

## Introduction

A robot swarm is robust to failure of individuals has always been a dominant view in swarm robotics research (e.g. (Bayindir and Şahin, 2007; Şahin et al., 2008)). It is expected that when an individual robot fails, the task left behind by the failed robot will be taken over by other robots in the swarm, and thus the swarm is robust. This view of a robust swarm has two underlying assumptions: the number of fault-free robots is significantly greater than the number of faulty robots, and that the failed robots do not interfere with other robots with respect to the operation of the swarm.

However, studies have demonstrated that the assumption that failed robots do not interfere does not hold for all modes of failure (Winfield and Nembrini, 2006). For a partly failed robot in which only some components are faulty whilst other components are still operational, the failed robot *can* and *will* interfere with the operation of the swarm. For example, in a swarm taxis scenario (swarm moving toward a beacon) investigated in Winfield and Nembrini (2006), a fault to the wheels while other components (e.g. wireless communication) are still operational causes physical anchoring of the robot swarm. Therefore, for such cases, there is a need to

handle these failures explicitly. One approach that is applicable for such cases is with explicit error detection and recovery. This approach consists of three stages: error detection, fault diagnosis, and recovery.

Error detection is a crucial first step as the activation of subsequent stages only occur when an error is detected. Previous studies on error detection in swarm robotics have looked at this problem for the case of a single faulty robot in the swarm. However, there is little work that directly addresses error detection when there are multiple (simultaneous) faulty robots in a swarm. In Christensen et al. (2009), the detection of faulty robots occur at the system-level for sensor faults that can be visibly detected by other robots. In Li and Parker (2009), fault detection is investigated for tightly-coupled multi-robot teams. In this paper, we investigate multiple faulty robots in the context of a foraging swarm robotic system in which the ability to forage for each robot can be affected by faults as well as the conditions in the operational environment.

Results from earlier work (Lau et al., 2011b,a) for a single faulty robot have demonstrated the potential of adaptive error detection with the collective self-detection (CoDe) scheme. In the CoDe scheme, a robot determines whether itself is faulty by cross-reference its behaviour with other robots within a logically defined neighbourhood. Each robot communicates (broadcast) its data to other robots in the same neighbourhood. The neighbourhood is defined by the communication range. If there are multiple failures in the same neighbourhood, the detection of faulty robots is harder and the CoDe scheme might work less effectively. This is because if there are more faulty robots in a neighbourhood, the CoDe scheme (which is analogous to a majority voting scheme) might (mis-)detect the fault-free robots as faulty, and vice versa. This is unwanted as the likelihood of multiple failures in a large swarm can be high (Carlson et al., 2004). Luckily, since the robots are mobile, the likelihood of a faulty robot in a neighbourhood of more faulty robots can be low. Besides, if the errors can be detected and recovered early, it can also reduce the likelihood of having multiple faulty robots in a neighbourhood.



The main contributions of this paper, therefore, are (1) an extended investigation on an immune-inspired CoDe scheme for multiple faulty robots in a swarm, (2) presentation on the calculation of the performance of detection for the case of multiple faulty robots, (3) an investigation on the correlation between the number of faulty robots and the required swarm size for effective error detection, and (4) the identification of the robot's communication range as a potential influencing factor on the performance of detection.

This paper is structured as follows. Section present the state-of-the-art on error detection in swarm robotics and the motivation of this paper. Section presents details on the set of experiments, and the experimental setup to investigate error detection for the case of multiple simultaneous faulty robots. Results from the experiments are presented in Section whilst Section concludes with the findings from the experiments in this paper.

## Background

There are many approaches for detecting errors in faulty robots. Generally, they can be grouped into model-driven and data-driven approaches. In model-driven approaches, analytical models of how a robot should behave are build and the actual behaviour is then compared to the predicted behaviour of the models. A problem with model-driven approaches is that the development of accurate models is often difficult, if not impossible, especially if the operational environment is not static or controlled (Christensen et al., 2007a). Due to the interactions between robots and the environment, as well as other natural factors, the state of the environment can change and this in turn can affect the behaviour of the robots.

Alternatively, a data-driven approach uses data produced during normal operation as the basis to infer the presence of a fault. This eliminates the need for precise analytical models of the robot's behaviour. In addition, it is possible to deploy the same robot swarm in many different environments. Therefore, data-driven approaches are generally more preferred.

Previous studies on data-driven error detection in swarm robotics have investigated scenarios of a single faulty robot in the swarm (e.g. (Canham et al., 2003; Christensen et al., 2007a,b; Lau et al., 2011b; Mokhtar et al., 2009)). However, having only one faulty robot in a robot swarm is a best-case scenario because the likelihood of multiple robots failing is high due to a variety of circumstances.

## From Single to Multiple Faulty Robots

The reason for the previous focus on a single faulty robot was that the proposed solution should scale to failure on multiple robots, as the detection is only based on data from one robot (e.g. (Canham et al., 2003; Christensen et al., 2007a; Mokhtar et al., 2009)). In other words, it is assumed that the changes in the behaviour of the robots are only

caused by faulty components. The environment in which the robot swarm operates has no impact on the behaviour of the robots. In this case, even with multiple faulty robots in the swarm, the detection of errors on each individual robot remained the same and unaffected by the number of faulty robots.

However, for many scenarios especially when the robot swarms are deployed in real-world environment, the operational environment does affect the behaviour of the robots. An example would be a robot foraging scenario in which the performance of foraging of each robot can not only be affected by the presence of faults but also by the amount of objects in the arena or the condition of the terrain. Therefore, instead of using data from a single robot, the CoDe scheme (Lau et al., 2011b,a) utilises data from a collective. The collective is defined over a logical neighbourhood based on the communication radius of an observer robot. In the CoDe scheme, the presence of a error is determined by cross-referencing a robot's behaviour with other robots within a neighbourhood. Results from the studies show that with the CoDe scheme, an adaptive error detection in the presence of time-varying environmental changes can be achieved.

As mentioned earlier, previous studies have only looked at error detection for a single faulty robot in the swarm. In practise, the probability of having multiple faulty robots in a swarm can be high. A survey on mobile robot failures in Carlson et al. (2004) found that the mean time between failures across all robot types surveyed is twenty four hours and the availability was fifty four percent. Indeed, the frequency of failure is very high.

To detect multiple faulty robots in a swarm is a challenging problem, in particular for scenarios in which the behaviour of the robots is affected by the operational environment as well as the presence of faults. First, detection approaches that operate on the basis of a single robot may not be applicable as changes in the environment that affect the behaviour may be detected as errors. This may lead to false positives. In Canham et al. (2003); Christensen et al. (2007a); Mokhtar et al. (2009), the robots are trained with a set of behaviour that is considered fault-free and thus the learning is static. During operation, changes in the environment that can affect the behaviour of the robots are not anticipated, and thus likely to cause the environmental changes to be detected as faults. Second, the assumption that there are more fault-free robots compared to faulty robots as employed in the CoDe scheme, might not be true for all scenarios. This is particularly in scenarios in which there are more faulty robots compared to fault-free robots in the neighbourhood. This leads to false negatives.

However, the fact that the robots are mobile minimises the frequency of such scenarios. The membership of robots in a logical neighbourhood is dynamic as the robots moves about in the arena. In addition, if errors can be detected early, the likelihood of having a neighbourhood with more

faulty robots can also be minimised. Therefore, this paper aims to investigate the following research questions

- Can the CoDe scheme be extended to multiple faulty robots?
- Is there any correlation between the swarm size and the number of faulty robots that can be detected with the CoDe scheme?

To investigate these research questions, the next two sections will provide details on the CoDe scheme, the evaluation metrics, and the set of experiments conducted.

## The Detection Framework

**The Detection Scheme** The detection of errors in faulty robots in this paper is based on the CoDe scheme proposed in Lau et al. (2011b,a). This scheme is analogous to a majority voting scheme (majority wins) which is the basis for social comparison when objective, non-social means are available (Festinger, 1954). However, the CoDe scheme is designed and implemented for self-detection of errors. Self-detection here means that instead of identifying whether other robots are faulty, it detects whether itself is faulty. This is inspired by the observed behaviour of self-isolation, to die alone, in ants. Instead of being actively located and isolated by healthy members, some species of ants infected by parasites tends to isolate themselves to die (Heinze and Walter, 2010). Taking this approach means that, at this stage, the detection can be more robust as it does not requires the identification of other robots in the swarm, as well as storing and keeping record of previous encounters with other robots. If the identification of faulty robots is required, it can be implemented on top of self-detection as proposed in Christensen et al. (2009). The pseudocode for the CoDe scheme is presented in Algorithm 1.

**The Classifier** The classifier used in this paper is the Receptor Density Algorithm (RDA) Owens et al. (2009) inspired by the T-cell receptor signalling mechanism in the immune system. By extraction of certain features of the generalised T-cell receptor, it was then mapped onto kernel density estimation. The RDA works as follow. The spectrum of input data is divided into  $s$  discretised location and a receptor  $\mathbf{x}_s$  is placed at each of these locations. The input data is the variable values from the robots used for the detection. A receptor has a length  $\ell = (\sqrt{2\pi})^{-1}$ , a position  $r_p \in [0, \ell]$ , and a negative feedback barrier  $\beta \in (0, \ell)$ . At each control cycle step  $t$ , each receptor takes input  $\mathbf{x}_i$  and performs a binary classification  $c_t \in \{0, 1\}$  to determine whether that location is considered anomalous. In general, the observation of one anomalous location is sufficiently representative to indicate the present of an anomaly at  $t$  (Owens et al., 2009).

The classification decision is determined by the dynamics of  $r_p$  and negative feedback  $r_n \in (0, \ell)$ . During training or

---

### Algorithm 1: Collective Self-Detection Scheme (CoDe)

---

**Input:** current data instance  $v$ , data instances from neighbours  $\mathcal{DN}$ , classifier  $\mathcal{A}$   
**Output:** report error  
**foreach** control cycle  $t$  **do**  
  **if**  $\text{CalculateNeighbour}(\mathcal{DN}) < 2$  **then**  
     $err = \mathcal{A}(v, \mathcal{DN}_{temp})$ ;  
    //  $\mathcal{DN}_{temp}$  is the data from previous control cycle having more than 2 neighbours  
  **else**  
     $err = \mathcal{A}(v, \mathcal{DN})$ ;  
     $\mathcal{DN}_{temp} = \mathcal{DN}$ ;  
    // assign current data from neighbours to  $\mathcal{DN}_{temp}$   
  **end**  
  **if**  $err$  **then**  
    Report  $err$ ;  
  **end**  
**end**

---

initialisation, after  $\mathbf{x}_i$  was presented, if the resulting  $r_p \geq \beta$  then a negative feedback  $r_n$  is generated which acts to reverse the progression of  $r_p$ . If  $r_p < \beta$ , no negative feedback will be generated,  $r_n = 0$ .

$$r_p(\mathbf{x}) = \sum_{i=1}^n \frac{1}{nh} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \quad (1)$$

$$r_n(\mathbf{x}) = \begin{cases} r_p(\mathbf{x}) - \beta, & \text{if } r_p(\mathbf{x}) \geq \beta \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The receptor position and negative feedback decay over time. During testing, for a new data instance  $\mathbf{v}$  if  $r_p^t > \ell$ , then the receptor generates an anomaly classification  $c_t = 1$ .

$$r_p^t(\mathbf{x}) = b \times r_p^{t-1}(\mathbf{x}) + gb \times K\left(\frac{\mathbf{x} - \mathbf{v}}{h}\right) - a \times r_n^{t-1}(\mathbf{x}) \quad (3)$$

where  $b \in \mathbb{R}^+$  is receptor position's decay rate,  $gb \in \mathbb{R}^+$  is current input stimulation rate,  $a \in \mathbb{R}^+$  is negative feedback's decay rate.

$$c(\mathbf{v}) = \begin{cases} 1, & \text{if } r_p^t(\mathbf{x}) \geq \ell \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

For the experiments in this paper, the  $K(\mathbf{x})$  in Eq. 1 and Eq. 3 is Gaussian kernel, the  $\beta = 0.01$ ,  $b = 0.02$ ,  $gb = 1.1$ , and  $a = 1.7$ .

## The Experiments

Two experiments are conducted: 1) an investigation of the potential of detecting errors in a swarm of multiple faulty robots with the CoDe scheme; 2) an investigation on the

correlation between the size of the swarm and the ability to detect errors.

**Fixed Swarm Size** This experiment investigates the robustness in detecting errors for multiple faulty robots with the CoDe scheme. The aim is to find out whether the approach that has been demonstrated to work well with a single faulty robot Lau et al. (2011b,a) can also be applied to the case of multiple faulty robots. More importantly, if the approach works with multiple failures then how will it degrade as the the number of failures increases.

In this experiment, faults are injected to robots in the system at 2500s (which is at control cycle 10). However, the fault models, number of faulty robots, and the duration for each fault are randomly generated. With a random number of faulty robots and fault durations, the number of faulty robots in a single simulation run can vary from one control cycle to another. For this reason, section describes how the performance is to be evaluated.

**Variable Swarm Size** This experiment investigates the possible correlation between the swarm size and the number of faulty robots that can be detected. This relates to the reliability of the error detection. The aim is to find a correlation  $n = ak + c$ , if exists, such that for  $k$  faulty robots there needs at least  $n$  robots in the swarm to ensure that the errors can be detected reliably with a true positive rate greater than or equals to  $x$ , a false positive rate of less than or equals to  $y$ , or both. For hardware redundancy, the number of redundant components generally suggested is  $2k + 1$  (Abd-El-Barr, 2006).

The configuration for the time of faults injection, number of faulty robots, duration for each fault in this experiment is the same as previous experiment. However, the swarm size is increased gradually starting from 10 robots. For every successive increment, an additional of two robot will be added.

## The Evaluation Metrics

The performance of detection is evaluated based on the performance is based on the true positive rate, false positive rate, and the (Latency), as in Lau et al. (2011a). Given:

- True Positive (TP) - an error is correctly classified;
- False Positive (FP) - a normal instance is incorrectly classified as an error;
- True Negative (TN) - a normal instance is correctly classified as normal; and
- False Negative (FN) - an error is incorrectly classified as a normal instance.

The True Positive Rate (TPR) is the proportion of the number of correctly classified errors over the total number of erroneous instances (Eq. 5).

$$TPR = \frac{TP}{TP + FN} \quad (5)$$

Similarly, the False Positive Rate (FPR) is the proportion of the number of incorrectly classified errors over the total number of normal instances (Eq. 6).

$$FPR = \frac{FP}{FP + TN} \quad (6)$$

The (Latency) metric evaluates how long the time has elapsed before an error is positively identified (Eq. 7). Given that  $t_{pd}$  is the fault detection time, and  $t_{ft}$  is the fault injection time, then

$$Latency = t_{pd} - t_{ft} \quad (7)$$

We present how TPR and FPR can be calculated for multiple faulty robots with reference to Figure 1. In the figure, there are ten robots in the system, labelled R1 to R10. In Figure 1(a), seven robots are faulty from time  $t=10$  onwards. However, the durations of faults between the robots are different as indicated by the black-coloured bar. For example, the fault in R1 lasts for 4 control cycles, from  $t=10$  to  $t=14$ . Since the durations faults are different, the number of faulty robots at each control cycle also differs. From  $t=10$  to  $t=11$ , there are seven faulty robots whereas from  $t=14$  to  $t=15$  there is only one faulty robot. By analysing the simulation data in this way, the TPR and FPR for each number of faulty robots can be calculated. For the scenario in Figure 1(a), there are instances for seven, six, five, three, and one faulty robot(s).

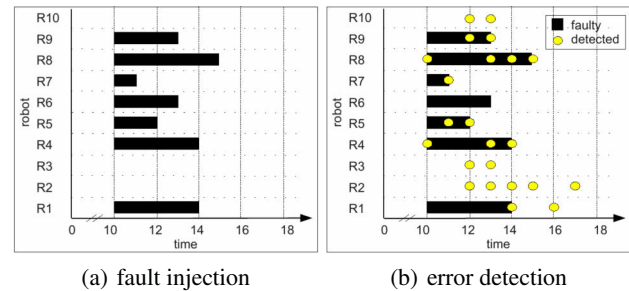


Figure 1: An illustration to show the calculation of TPR and FPR given information regarding the fault injection time, duration of fault, and detected errors.

In Figure 1(b), the circles represent the instances of error being detected. It can be seen that there are many instances of false positive (detection of error even when no fault was injected), e.g. with robots R2, R3, and R10. To calculate the TPR and the FPR, starts at  $t=11$ . At  $t=11$ , there are seven faulty robots. Therefore, the TPR for the case of seven faulty robots is  $2/7$ . Similarly, the FPR is  $0/3$ . At  $t=12$ , the TPR for six faulty robots is  $2/6$  whilst the FPR is  $3/4$ , and so on.

## Experimental Setup

The experiments in this paper were carried out in simulation. The context of the work is a robot swarm in a foraging scenario. The source code for the foraging swarm robotic system, data, and scripts used to produce results for this paper are available at <http://sites.google.com/site/huikenglau/shared>.

### Simulation Settings

The software used to implement the foraging robot swarm is the Stage plug-in (Gerkey et al., 2003). The robot swarm is placed in an arena to continuously locate, transport, and deposit objects until the end of simulation. At any time, a random number of robots can fail.

**Arena** The arena is an octagonal-shaped area of  $12\text{m} \times 12\text{m}$  with a circular base of  $3\text{m}$  in diameter in the centre. Objects are placed at random<sup>1</sup> locations but outside of the base at a default object-replenishing-rate (OPR) of 0.10. This means that the probability of adding an object at every second is 0.10.

**Robot** The physical robot from which the simulation model was based is the Linuxbot from Bristol Robotics Laboratory<sup>2</sup>. Each robot is equipped with an array of sensors and components needed for foraging. The default moving speed of the robots is  $0.15 \text{ m.s}^{-1}$  with a communication range of  $2\text{m}$ .

**Fault** Each robot in the swarm is subject to a particular fault in the wheels whilst other components are still functioning, as examined in Winfield and Nembrini (2006). Three models of faulty wheels were simulated: complete  $P_{CP}$ , partial  $P_{PT}$ , and gradual  $P_{GR}$ . The  $P_{CP}$  causes the wheels of a robot to stop responding completely and thus the robot is unable to proceed with foraging. With  $P_{PT}$ , the robot moves with a reduced speed and thus resulting in less objects being collected when compared to a fault-free condition. With  $P_{GR}$ , the robot moves with a gradually reducing speed until eventually it comes to a complete stop. In simulation,  $P_{CP}$  is simulated by setting the left wheel to left turn by  $10^\circ$  causing the robot to move in circle. For  $P_{PT}$ , the robots move with a reduced speed of  $0.45 \times 10^{-1} \text{ ms}^{-1}$  whilst for the  $P_{GR}$  the speed of the robot is reduced by  $0.10 \times 10^{-3} \text{ ms}^{-2}$ . The mode of the fault is transient; the fault lasts for a random period of time and then the robot recovers and continue with normal operation.

**Environment** Two different scenarios in which the robot swarm operates: constant OPR (CST), varying OPR ( $V_{OPR}$ ). In a CST scenario, the OPR is fixed at 0.10. On the other

hand, in a  $V_{OPR}$  scenario, the OPR alternates between 0.10 to 0.01 at different intervals.

**A Simulation Run** A simulation starts with 100 initial objects placed randomly in the arena. A maximum number of objects in the arena at any one time is capped at 400 units to avoid overcrowding. Each object is a small red coloured square box which can be sensed by the camera on each robot and picked up by the grippers on the robot. Robots depart from the base and the heading for each robot  $R_i$  is based on the formula  $R_i = \frac{i \times 2\pi}{n}$ ,  $n$  is the number of robots in the swarm. The robots will continuously carry out foraging until the end of the simulation. In this paper, each simulation lasts for 10,000s. Periodically (i.e. every control cycle, 250s), data on the number of objects collected `obj`, energy used `eng`, and distance travelled `dist` for each robot are extracted and output as csv files. For each variable, an instance of the CoDe scheme is executed separately and an error is considered detected if it is reported in at least one of the variables. The  $h$  in Eq. 1 for `obj` = 1.0, `eng` = 12.0, and `dist` = 2.5

## Experimental Results

### Fixed Swarm Size

Figure 2 is the result for the TPR and FPR in detecting errors for different number of faulty robots in a CST scenario. Each point in the graphs represent the TPR or FPR calculated over 100 repeated runs. Note that since the number of faulty robots and the duration of each fault are random, the number of instances for each group of faulty robots also differs. For example, out of the 100 runs, there are 1208 instances of eight faulty robots, 777 instances for seven faulty robots, 948 instances of six faulty robots and so on.

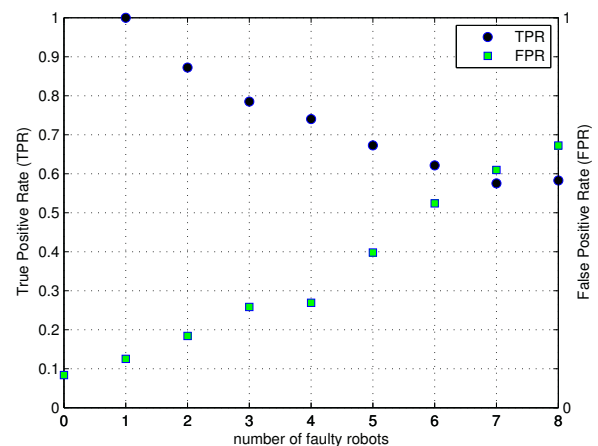


Figure 2: The TPR and the FPR for detecting multiple faulty robots in a CST scenario with the CoDe scheme.

In Figure 2, the results show that as the number of faulty

<sup>1</sup>The random number generator used is from GSL-GNU Scientific Library.

<sup>2</sup><http://www.ias.uwe.ac.uk/Robots/linuxbot.htm>



robots increases, the performance of detection decreases. Note that in this experiment, no recovery is included. Therefore, a TPR of 1.00 with one faulty robot means that the errors in that faulty robot can always be detected. Similarly, a TPR of 0.55 with eight faulty robots means that there is 55% chance that the errors in all eight faulty robots will be detected. This is possible because there are still two fault-free robots for the faulty robots to cross-referencing their data. In addition, because of the dynamic neighbourhood of the robots from one control cycle to another, a faulty robot might also be cross-referencing its data against others from the previous control cycle (refer line 5 in Algorithm 1).

The results show that with every increase of one faulty robot, the TPR decreases at an approximately constant rate of 0.10. This means the probability of detecting errors in all faulty robots decreases as the number of fault-free robots decreases. This is expected because the likelihood of more than one faulty robot in the logical neighbourhood is increased, and thus resulting in more false negatives. Similarly, the increase of the number of faulty robots also increases the FPR. The increase in the FPR is also approximately 0.10 for each addition of one faulty robot. Due to the same reason for the TPR, each addition of faulty robot increases the likelihood of fault-free robots to classify itself as faulty (false positives).

The result for the Latency in detecting the errors is shown in Figure 3. From the 100 runs, the median Latency is 1 control cycle. This means that the errors are detected in the next control cycle after faults were injected. This is a positive result because if recovery measures were implemented, the number of multiple simultaneous faulty robots can be reduced.

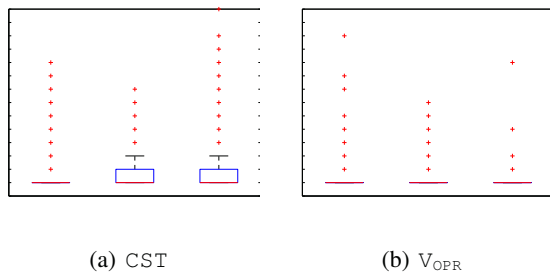


Figure 3: The Latency in detecting the models of fault of the wheels in a CST and a  $V_{OPR}$  scenario. The '+' points on the boxplots are the outliers.

The graphs in Figure 4 compares the performance between a CST and a  $V_{OPR}$  scenario. Overall, there is no significant difference between the performance in the TPR and the FPR. The drop in the TPR and the increase in the FPR as the number of fault robots increases are similar to the results for a CST scenario. In fact, some of the results for the  $V_{OPR}$  scenario are better than the CST scenario. This

is because even with a fixed OPR, the presence of multiple faulty robots can significantly affect the ability to positively identify errors. Nevertheless, this result shows that CoDe scheme works well in a non-dynamic as well as a dynamic environment even with multiple faulty robots in the swarm. This is encouraging as it further supports that the CoDe scheme can be adaptive to dynamic environments.

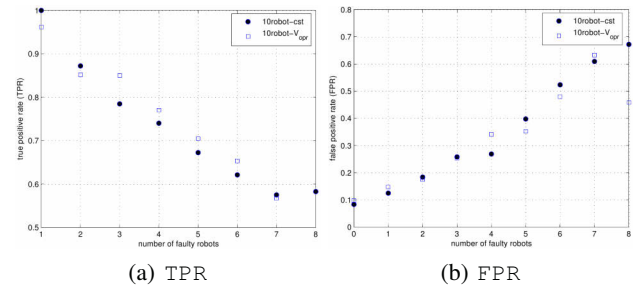


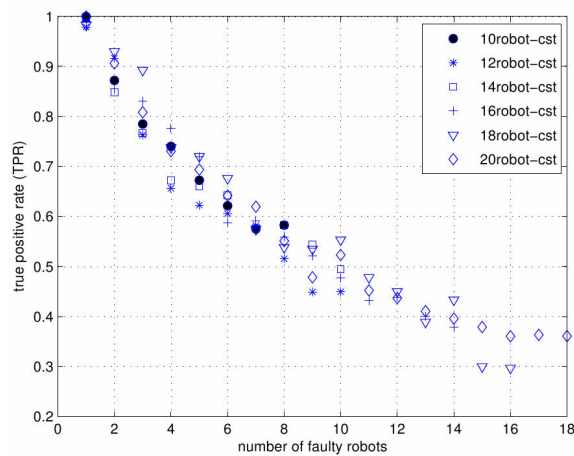
Figure 4: The TPR and FPR in detecting errors for a robot swarm with multiple faulty robots in a CST and a  $V_{OPR}$  scenario.

### Variable Swarm Size

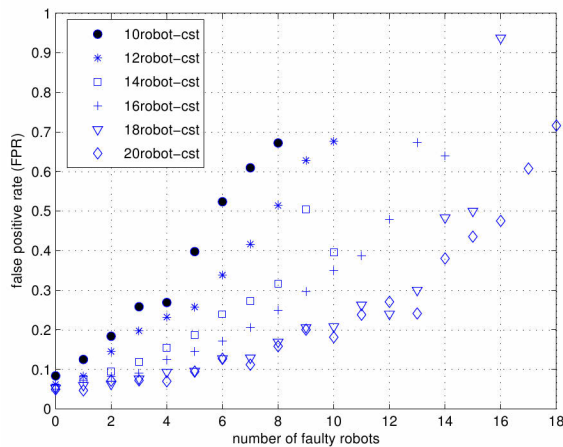
Figure 5 show the TPR and FPR for multiple faulty robots with different swarm sizes. A general observation is that as the swarm size increases, the performance of detection also improves. For example, on the x-axis with two faulty robots in Figure 5(a), as the swarm size is increased from 10 robots to 18 robots, the TPR also increases (from about 0.85 to slightly above 0.90). Similarly in Figure 5(b) with two faulty robots, as the swarm size increases from 10 to 18 robots the FPR decreases from about 0.20 to less than 0.10.

However, note that the increase in the TPR does not occur in all cases. In some cases, rather counter-intuitive. For example with eight faulty robots in Figure 5(a), the TPR decreases from slightly below 0.60 to only above 0.50 when the swarm size increases. This observation is interesting and worth further investigation. One particular factor comes to mind is the communication range of each robot. This parameter influences the size of the logical neighbourhood. Here, it is set to 2m radius. From the results, it appears that an increase in the swarm size does not guarantee an increase in the neighbourhood size (i.e. the number of robots in the neighbourhood) at each control cycle. Therefore, this aspect will be investigated in the near future.

From this result of varying swarm size, the required swarm size for different number of faulty robots can be calculated. For example, in order to not falsely detect errors at 80% of the time in a swarm with four or less faulty robots (i.e.  $FPR = 0.80$ ) the swarm needs to have at least 12 robots. Similarly, with six or less faulty robots, a swarm of at least 16 robots is needed. From this trend, it seems that a swarm of  $n = k + 10$  is required to achieve a FPR less than 0.20 with  $k$  faulty robots.



(a) TPR



(b) FPR

Figure 5: The TPR and FPR in detecting errors for different swarm sizes.

Based on the same analysis, to be able to detect four faulty robots or less at 80% of the time (i.e.  $TPR = 0.80$ ), the swarm needs to have more than 20 robots. Unlike the TPR, based on current results, it is impossible and unrealistic to extrapolate the required swarm size with more than four faulty robots.

A general observation is that the swarm size required is greater than  $2k + 1$ . This is comparable to the generally used hardware redundancy Abd-El-Barr (2006). The reason is that in swarm robotics the robots are mobile and there is no guarantee that for  $k$  faulty robots, there will be at least  $2k + 1$  fault-free robots in the same logical neighbourhood. This observation hints that there are other factors involved and one particular parameter that came to mind is the communication range of the robots. For confirmation, this parameter will be investigated in the near future.

### Comparison with Q-test

The performance of error detection with multiple faulty robots using the CoDe scheme with the RDA is compared

with the Q-test (Gibbons, 1994) (Table 1, Table 2). Dixon's Q test (Gibbons, 1994), or simply the Q-test, is a non-parametric technique that can be used for error detection. It has been applied for error detection in the case of a single faulty robot in Lau et al. (2011a) and shown to produced the best results when compared to other statistical classifiers such as T-test, Quartile-based, and Extreme Studentised Deviate.

In Table 1, the RDA has consistently achieved a higher TPR when compared to the Q-test from all swarm sizes (from 10 to 20). However, a similar result is not obtained for the FPR (Table 2). When the number of faulty robots increases the FPR for the RDA increases. From the perspective of the CoDe scheme (i.e. majority voting), this is expected, in particular when the number of fault-free robots is significantly less than the number of faulty robots. Having said that, a more detailed investigation will be conducted in the near future.

Table 1: The difference of the TPR of the RDA and the Q-test (i.e. RDA-Q-test) in detecting errors with multiple faulty robots. Note that for the TPR, a positive value means a better result.

| No. Faulty robots | 10   | 12   | 14   | 16   | 18   | 20   |
|-------------------|------|------|------|------|------|------|
| 1                 | 0.18 | 0.13 | 0.15 | 0.15 | 0.09 | 0.14 |
| 2                 | 0.27 | 0.18 | 0.26 | 0.19 | 0.23 | 0.15 |
| 3                 | 0.28 | 0.25 | 0.31 | 0.26 | 0.22 | 0.25 |
| 4                 | 0.33 | 0.30 | 0.32 | 0.29 | 0.27 | 0.22 |
| 5                 | 0.36 | 0.28 | 0.34 | 0.32 | 0.27 | 0.28 |
| 6                 | 0.33 | 0.28 | 0.30 | 0.26 | 0.30 | 0.27 |
| 7                 | 0.21 | 0.28 | 0.30 | 0.31 | 0.21 | 0.31 |
| 8                 | 0.20 | 0.27 | 0.30 | 0.29 | 0.27 | 0.30 |
| 9                 |      | 0.22 | 0.26 | 0.29 | 0.27 | 0.28 |
| 10                |      | 0.12 | 0.26 | 0.27 | 0.26 | 0.28 |
| 11                |      |      | 0.22 | 0.22 | 0.27 | 0.24 |
| 12                |      |      | 0.13 | 0.21 | 0.22 | 0.23 |
| 13                |      |      |      | 0.19 | 0.19 | 0.22 |
| 14                |      |      |      | 0.12 | 0.20 | 0.20 |
| 15                |      |      |      |      | 0.12 | 0.19 |
| 16                |      |      |      |      | 0.02 | 0.18 |
| 17                |      |      |      |      |      | 0.15 |
| 18                |      |      |      |      |      | 0.09 |

### Conclusion

We have presented our initial investigation on error detection for multiple faulty robots in the swarm. Specifically, we looked at scenarios in which the behaviour of the robots can be affected by both faulty components and changes in the environment. In addition, the way to calculate the performance metrics, namely the true positive rate, false positive rate, and latency for the case of multiple faulty robots are also presented. Revisiting the research questions, results from the first experiment give evidence that the CoDe scheme, which work for a single faulty robot, performs well for multiple faulty robots. In the second experiment, the general results show that as the swarm size is increased, the performance in detecting errors with multiple faulty robots also increases. In particular, the size of the swarm needs to be greater than

Table 2: The difference of the FPR of the RDA and the Q-test (e.e RDA-Q-test) in detecting errors with multiple faulty robots. Note that for the FPR, a negative value means a better result.

| No. Faulty Robots | 10    | 12    | 14    | 16    | 18    | 20    |
|-------------------|-------|-------|-------|-------|-------|-------|
| 1                 | -0.04 | -0.08 | -0.09 | -0.10 | -0.09 | -0.08 |
| 2                 | 0.01  | -0.01 | -0.02 | -0.07 | -0.08 | -0.07 |
| 3                 | 0.07  | 0.04  | -0.05 | -0.06 | -0.07 | -0.06 |
| 4                 | 0.07  | 0.05  | -0.01 | -0.02 | -0.07 | -0.07 |
| 5                 | 0.16  | 0.07  | 0.03  | 0.01  | -0.06 | -0.06 |
| 6                 | 0.19  | 0.08  | 0.02  | -0.03 | -0.04 | -0.01 |
| 7                 | 0.27  | 0.09  | 0.05  | 0.00  | -0.07 | -0.03 |
| 8                 | 0.30  | 0.16  | 0.13  | 0.04  | -0.02 | 0.02  |
| 9                 |       | 0.17  | 0.11  | 0.01  | -0.02 | 0.02  |
| 10                |       | 0.27  | 0.14  | 0.08  | 0.00  | 0.01  |
| 11                |       |       | 0.24  | 0.06  | 0.03  | 0.03  |
| 12                |       |       | 0.26  | 0.11  | 0.00  | 0.04  |
| 13                |       |       |       | 0.19  | -0.08 | -0.02 |
| 14                |       |       |       | 0.24  | 0.13  | 0.08  |
| 15                |       |       |       |       | 0.25  | 0.10  |
| 16                |       |       |       |       | 0.38  | 0.11  |
| 17                |       |       |       |       |       | 0.15  |
| 18                |       |       |       |       |       | 0.31  |

$2k + 1$  where  $k$  is the number of faulty robots. The results also suggest the need for further investigation on the correlations between swarm size, communication radius, and the performance of detection.

### Acknowledgements

We acknowledge the Swedish Foundation for Strategic Research (SSF) SYNOPSIS Project and Artificial Intelligence Research Unit, Universiti Malaysia Sabah for supporting this work.

### References

- Abd-El-Barr, M. (2006). *Design And Analysis of Reliable And Fault-Tolerant Computer Systems*. Imperial College Press.
- Bayindir, L. and Şahin, E. (2007). A Review of Studies in Swarm Robotics. *Turkish Journal on Electrical Engineering and Computer Sciences*, 15(2):115–147.
- Canham, R., Jackson, A., and Tyrrell, A. (2003). Robot Error Detection Using an Artificial Immune System. In *Proce. 2003 NASA/DoD Conf. Evolvable Hardware*, pages 199–207. IEEE Computer Society.
- Carlson, J., Murphy, R., and Nelson, A. (2004). Follow-up Analysis of Mobile robot Failures. In *Proc. IEEE Intl. Conf. Robotics and Automation*, pages 4987–4994. IEEE.
- Christensen, A., O’Grady, R., Birattari, M., and Dorigo, M. (2007a). Automatic Synthesis of Fault Detection Modules for Mobile Robots. In *Proc. 2nd NASA/ESA Conf. Adaptive Hardware and Systems*, pages 693–700. IEEE Computer Society.
- Christensen, A., O’Grady, R., Birattari, M., and Dorigo, M. (2007b). Exogenous fault detection in a collective robotic task. In *Proc. 9th European Conf. Advances in Artificial Life*, LNAI 4648, pages 555–564. Springer.
- Christensen, A. L., O’Grady, R., and Dorigo, M. (2009). From Fireflies to Fault-Tolerant Swarms of Robots. *IEEE Trans. Evolutionary Computation*, 13:754–766.
- Şahin, E., Girgin, S., Bayindir, L., and Turgut, A. (2008). Swarm robotics. In Blum, C. and Merkle, D., editors, *Swarm Intelligence: Introduction and Applications*, pages 87–100. Springer.
- Festinger, L. (1954). A Theory of Social Comparison Processes. *Human Relations*, 7(2):117–140.
- Gerkey, B., Vaughan, R. T., and Howard, A. (2003). The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In *Proc. 11th Intl. Conf. Advanced Robotics*, pages 317–323.
- Gibbons, R. D. (1994). *Statistical Methods for Groundwater Monitoring*. John Wiley & Sons, Inc.
- Heinze, J. and Walter, B. (2010). Moribund Ants Leave Their Nests to Die in Social Isolation. *Current Biology*, 20:249–252.
- Lau, H., Bate, I., Cairns, P., and Timmis, J. (2011a). Adaptive Data-Driven Error Detection in Swarm Robotics with Statistical Classifiers. *Robotics and Autonomous Systems*, 59(12):1021–1035.
- Lau, H., Timmis, J., and Bate, I. (2011b). Collective Self-detection Scheme for Adaptive Error Detection in a Foraging Swarm of Robots. In *Proc. ICARIS 2011*, LNCS 6825, pages 254–267. Springer.
- Li, X. and Parker, L. (2009). Distributed sensor analysis for fault detection in tightly-coupled multi-robot team tasks. In *IEEE Intl. Conf. Robotics and Automation*, pages 3103–3110.
- Mokhtar, M., Timmis, J., Tyrrell, A., and Bi, R. (2009). A Modified Dendritic Cell Algorithm for On-Line Error Detection in Robotic System. In *Proc. CEC2009*, pages 2055–2062. IEEE Press.
- Owens, N., Greensted, A., Timmis, J., and Tyrell, A. (2009). T Cell Receptor Signalling Inspired Kernel Density Estimation and Anomaly Detection. In *Proc. ICARIS 2009*, LNCS 5666, pages 122–135. Springer.
- Winfield, A. and Nembrini, J. (2006). Safety in Numbers: Fault Tolerance in Robot Swarms. *Intl. J. Modelling Identification and Control*, 1(1):30–37.

## The use of Agent-based Simulation to Discover Extreme Cases in Immune-Interactions with Early-Stage Cancer Scenarios

Grazziela Figueredo<sup>1</sup>, Peer-Olaf Siebers<sup>1</sup>, Douglas Augusto<sup>2</sup>, Helio Barbosa<sup>2</sup> and Uwe Aickelin<sup>1</sup>

<sup>1</sup>Intelligent Modelling and Analysis Research Group, School of Computer Science,  
The University of Nottingham, NG8 1BB, UK

<sup>2</sup>Laboratório Nacional de Computação Científica-MCT, Petrópolis, Brazil  
gzf@cs.nott.ac.uk, pos@cs.nott.ac.uk, douglas@lncc.br, hcbm@lncc.br, uxa@cs.nott.ac.uk

Early-stage cancer and its interactions with the immune system are still not fully understood. In order to better understand these processes, researchers employ different methods. Simulation and in particular, agent-based simulation (ABS) have been found useful tools for understanding it (Look et al., 1981; Castiglione et al., 1999, 2001; Bonabeau, 2002; Figueredo and Aickelin, 2011; Figueredo et al., 2013a,b).

In a previous study (Figueredo et al., 2013b) we have built an ABS model to study the interplay of immune cells and early-stage cancer. The model considers interactions between tumour cells and immune effector cells, as well as the immune-stimulatory and suppressive cytokines IL-2 and TGF- $\beta$ . IL-2 molecules mediate the immune response towards tumour cells. They interfere on the proliferation of effector cells according to the number of tumour cells in the system. Conversely, TGF- $\beta$  stimulates tumour growth and suppresses the immune responses by inhibiting the activation of effector cells and reducing tumour-antigen expression.

In order to validate our model, we used a well-established mathematical model found in the literature (Arciero et al., 2004). While at average both models do not show a statistical significant difference, some additional trends in the results of the ABS model are observed. As ABS is a stochastic simulation method, it was run for multiple times. Instead of having one solution, as it is the case for a deterministic mathematical model, ABS produces a variety of outcomes. These solutions are usually very similar. In our cases study, however, we could observe some instances which could not have been observed by using analytical methods (see Figure 1).

The use of ABS modelling has therefore led to the discovery of additional “rare” patterns, which we would have not been able to derive by using analytical methods. These “extreme cases” indicate that there might be circumstances where the tumour cells are completely eliminated by the immune system, without the need of any cancer therapies. We strongly believe that the observed emergent behaviour produced by stochastic simulation can make a useful con-

tribution to assisting immunological research. With the additional information supplied from the ABS, immunologists can test new hypotheses and further investigate whether these extreme cases actually occur in reality and why.

Currently, we are working on a methodology for defining experimental conditions that would allow us to observe similar emergent behaviour in other simulation experiments related to early-stage cancer research. One important aspect here is to investigate the statistical conditions under which emergent behaviour starts to appear. The questions we are looking at are:

1. How many replications of our stochastic simulation do we have to run before we can expect to see rare behaviours?
2. Is there any regularity in the growth of these rare emerging patterns?
3. What are the factors that need to be considered when predicting the occurrences of emerging patterns (e.g. level of dynamics in the model)?

The patterns obtained in our previous work were a result of 50 independent runs of the ABS model (Figueredo et al., 2013b). In order to further advance our knowledge regarding these patterns we are currently running experiments with 10,000 independent runs in order to verify whether there is any regularity in pattern growth. We also intend to validate our results with immunologists. It is hoped that the development of a methodology to further investigate extreme cases could assist in defining suitable vaccination strategies and the appropriateness of cancer treatments by the prediction of the possible outcome scenarios and how frequently they take place.

### Acknowledgements

This work was supported by the Advanced Data Analysis Centre (ADAC) at the University of Nottingham. The research leading to these results has also received funding from Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and FAPERJ (grant E-26/102.025/2009).



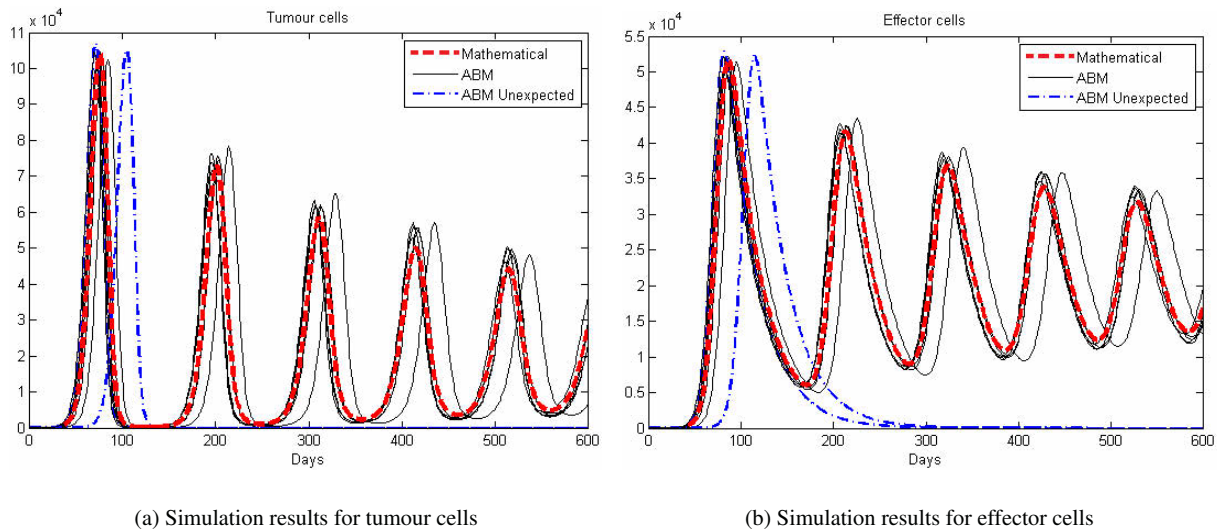


Figure 1: Simulation results: the dashed line (red) shows the mathematical output; the lines in black show exemplar ABS results for 6 runs. As it can be seen, there are some results very close to the mathematical formulation and others presenting more variability due to the ABS stochastic behaviour. These variations, however, follow the same pathway as the analytical solution. The dashed-dotted line (blue) shows the rare cases determined by the ABS simulations.

## References

- Arciero, J. C., Jackson, T. L., and Kirschner, D. E. (2004). A mathematical model of tumor-immune evasion and siRNA treatment. *Discrete and continuous dynamical systems - series B*, 4(1):39–58.
- Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 99, pages 7280–7287.
- Castiglione, F., Mannella, G., Motta, S., and Nicosia, G. (1999). A network of cellular automata for the simulation of the immune system. *Journal of Modern Physics C – Physics and Computer*, 10(4):677–686.
- Castiglione, F., Motta, S., and Nicosia, G. (2001). Pattern recognition by primary and secondary response of an artificial immune system. *Theory in Biosciences*, 120(2):93–106.
- Figueredo, G. P. and Aickelin, U. (2011). Comparing system dynamics and agent-based simulation for tumour growth and its interactions with effector cells. In *Proceedings of the International Summer Computer Simulation Conference*, pages 15–22.
- Figueredo, G. P., Joshi, T. V., Osborne, J. M., Byrne, H. M., and Owen, M. R. (2013a). On-lattice agent-based simulation of populations of cells within the open-source chaste framework. *Interface Focus*, 2(3).
- Figueredo, G. P., Siebers, P.-O., and Aickelin, U. (2013b). Investigating mathematical models of immuno-interactions with early-stage cancer under an agent-based modelling perspective. *BMC Bioinformatics*, 14(6).
- Look, A. T., Schriber, T. J., Nawrocki, J. F., and Murphy, W. H. (1981). Computer simulation of the cellular immune response to malignant lymphoid cells: logic of approach, model design and laboratory verification. *Immunology*, 43(4):677–690.

# Learning to Solve Bin Packing Problems with an Immune Inspired Hyper-heuristic

Kevin Sim<sup>1</sup>, Emma Hart<sup>1</sup> and Ben Paechter<sup>1</sup>

<sup>1</sup> Institute for Informatics and Digital Innovation,  
Edinburgh Napier University,  
Merchiston Campus, Edinburgh, EH10 5DT  
k.sim@napier.ac.uk

## Abstract

Motivated by the natural immune system's ability to defend the body by generating and maintaining a repertoire of antibodies that collectively cover the potential pathogen space, we describe an artificial system that discovers and maintains a repertoire of *heuristics* that collectively provide methods for solving *problems* within a problem space. Using bin-packing as an example domain, the system continuously generates novel heuristics represented using a tree-structure. A novel affinity measure provides stimulation between heuristics that cooperate by solving problems in different parts of the space. Using a test suite comprising of 1370 problem instances, we show that the system self-organises to a minimal repertoire of heuristics that provide equivalent performance on the test set to state-of-the-art methods in hyper-heuristics. Moreover, the system is shown to be highly responsive and adaptive: it rapidly incorporates new heuristics both when entirely new sets of problem instances are introduced or when the problems presented change gradually over time.

## Introduction

Heuristic search methods have been shown to be successful in solving a wide-range of real-world problems. Typically for a given application domain, a range of heuristics for solving problems will exist; these might range in nature from simple rules encapsulating expert knowledge to complex search algorithms that need to be tuned by experts to work. Commonly, different heuristics will work well on problems in different parts of the *problem space*. By collecting together a set of heuristics, it is hoped that collectively, the weaknesses of individual heuristics can be compensated for by other heuristics in the set (Burke et al., 2003). The goal of the *hyper-heuristics* field is to find automated methods that can both generate appropriate sets of heuristics and provide a means of selecting between heuristics in the set, given either a new problem instance or even a partially solved problem instance.

While such approaches have proved successful in many application areas, most hyper-heuristic approaches fail to *continuously* learn from experience. On the one hand, the failure to exploit previous knowledge leads to inefficient hyper-heuristics; on the other, if the characteristics of instances of problems in the domain change over time, a

hyper-heuristic may need to be completely re-tuned or in the worst case redesigned periodically. An 'ideal' hyper-heuristic would be able to exploit previous knowledge through access to some kind of memory, rapidly adapt existing knowledge to new circumstances, and additionally, generate new knowledge when previous knowledge is not applicable.

We observe that the immune system fulfils very similar properties in its role as a host maintenance and defence system. The immune system maintains a repertoire of antibodies that has been shown theoretically to cover the space of potential pathogens. Clonal selection mechanisms provide a means of rapidly adapting existing antibodies to new variants of previous pathogens; meta-dynamic processes are able to generate novel antibodies; a memory mechanism enables the immune system to respond rapidly when faced with pathogens it has previously been exposed to.

Using this analogy, we present a system that is shown experimentally to outperform single human-designed heuristics by a significant margin and furthermore, is shown to be more adaptable and responsive than a recent state-of-the-art hyper-heuristic approach when faced with a continually changing problem landscape, thereby addressing the needs of real-world practitioners more fully. The novel system described has the following features:

- it generates novel heuristics from a library of component parts
- it utilises meta-dynamic processes to both add and remove heuristics from the system resulting in a self-organising network of heuristics
- it sustains a network of interacting heuristics of minimal size that collectively solve problems from the whole of the problem space
- it encapsulates memory in the sustained network enabling rapid adaptation to new problems

## Background

We briefly cover some background in relation to the tested application domain of bin-packing and outline the immunol-

ogy that inspired this approach before describing the system in detail.

## Immunology

Artificial Immune Systems (AIS) algorithms have been applied in many domains, including solving combinatorial optimisation (CO) problems (Kromer et al., 2012). Unlike other biologically inspired paradigms there is no de-facto model used by AIS practitioners. Of the models used the one most frequently applied to CO problems is *clonal selection theory* (Burnet, 1959), however, many other paradigms exist: we exploit the *idiotypic network* (Jerne, 1974) model in this work, in part due to its plasticity and its ability to describe the memory mechanism exhibited by the immune system.

Idiotypic network inspired models have been developed to address machine learning problems such as clustering (Neal, 2003), however the most relevant line of work in relation the model proposed in this article is in the robotics domain. An idiotypic network model to perform behaviour arbitration in mobile robots was one of the first applications in AIS (Watanabe et al., 1998) and spawned subsequent related work. In these early approaches, antibodies in the network represented behaviours (actions); antibodies were stimulated by environmental conditions and further suppressed or stimulated by interactions with other antibodies in a modified version of Farmer's original equation (Farmer et al., 1986). The immune repertoire consisted of a set of antibodies which collectively covered a space of appropriate actions required to achieve the robot goals in the space defined by environment. In this early work, antibodies were pre-defined; research focused on evolving connections and matching strengths between antibodies. In more recent work, Whitbrook *et al* have extended earlier work by using an evolutionary algorithm in separate learning phase to produce antibodies which are used to seed the network. The benefits of seeding the network with a diverse and novel set of antibodies are described in Whitbrook et al. (2010).

In this paper, we adopt a similar approach to Whitbrook *et al* in recognising the need to develop a diverse set of antibodies for potential inclusion in the network. In contrast to previous work however, our mechanism is not just used to seed the network in an initial phase but continuously generates a stream of potential antibodies which are either incorporated into the network or rejected according to a meta-dynamic process. This results in a learning scheme in which the network is able to continuously adapt over time.

## HyperHeuristics

Originally described as “*heuristics to select heuristics*” (Burke et al., 2003) the field of hyper-heuristics has evolved to also encompass “*heuristics to generate heuristics*” (Burke et al., 2010); both methods have the common goal of searching a landscape defined by the heuristics in order to find a procedure for solving a problem, rather than

searching directly over the solution space defined by the problem itself. Hyper-heuristic methods have been widely applied to bin-packing problems. We discuss the most relevant research briefly below.

**Heuristic generation** Genetic Programming (GP) is typically used a method of generating new heuristics. In Burke et al. (2006, 2012) GP was used to automate the design of heuristics for the bin-packing problem in multiple dimensions. Using a small set of benchmarks, they found the generated heuristics to be competitive with human-designed heuristics. In Sim and Hart (2013), the authors introduced Single Node Genetic Programming as a method to evolve new heuristics for bin-packing. SNGP, introduced in Jackson (2012), differs from the conventional GP model introduced by Koza (1992) in a number of key respects. A single tree structure is used to represent a population of possible trees by allowing any node to be the start node. Only mutation is used to change connections with the tree, alleviating the undesirable affect of *bloat* found in conventional GP and enabling different network structures to emerge in addition to classical tree structures.

**Heuristic selection** Most hyper-heuristics use a *fixed* set of low-level heuristics (whether generated or hand-built) and define or learn a model to select an appropriate heuristic based on the current state of the problem or a description of the problem characteristics. In contrast, Sim and Hart (2013) introduce an island model based on the concept of *Cooperative Coevolution* (Potter and De Jong, 2000) in which a set of islands each contribute a heuristic to a collaborating set that collectively are able to solve a problem. Crucially, the number of islands is not prefixed but is adaptable. In Sim and Hart (2013), an island contains a population described using SNGP; each island contributes its best heuristic to the collaboration set. Based on a set of 685 test instances, the authors showed that the model was able to solve more instances and reduce the number of extra bins required than an equivalent sized set of man-made heuristics from the literature.

In the current work, a simplified version of SNGP is used to generate novel heuristics and the island model described above is replaced by an AIS algorithm which is shown to be able to find, maintain and adapt a collaborating set of heuristics that equals or outperforms other previous approaches.

## 1D BPP and Benchmarks

The objective of the one dimensional bin packing problem (BPP) is to find a packing which minimises the number of containers,  $b$ , of fixed capacity  $c$  required to accommodate a set of  $n$  items with weights  $\omega_j : j \in \{1 \dots n\}$  falling in the range  $1 \leq \omega_j \leq c, \omega_j \in \mathbb{Z}$  whilst enforcing the constraint that the sum of weights in any bin does not exceed the bin capacity  $c$ . The lower and upper bounds on  $b$ , ( $b_l$  and

$b_u$ ) respectively, are given by equation 1. Any heuristic that does not return empty bins will produce, for a given problem instance,  $p$ , a solution using  $b_p$  bins where  $b_l \leq b_p \leq b_u$ .

$$b_l = \left\lceil \sum_{j=1}^n \omega_j \div c \right\rceil, \quad b_u = n \quad (1)$$

Table 1 shows the parameters from which the benchmark problem instances used in this study were generated. Data sets *ds1*, *ds2* & *ds3*, introduced by Scholl et al. (1997) all have optimal solutions that vary from the lower bound given by Equation 1. However all are known and have been solved since their introduction (Schwerin and Wäscher, 1997). All of the instances from *FalU* and *FalT*, introduced by Falkenauer (1996), have optimal solutions at the lower bound except for one (Gent, 1998).

Six deterministic heuristics commonly cited in the literature are used for comparison — FFD, DJD, DJT, ADJD, BFD and SS. Descriptions of each of these heuristics can be found in Sim and Hart (2013). Note that in the implementation used here, each deterministic heuristic is presented with each problem instance's items pre-sorted in descending order of size.

## Implementation

The system comprises of three main parts: a database of problem instances, a heuristic generator and the AIS as illustrated by Figure 1.

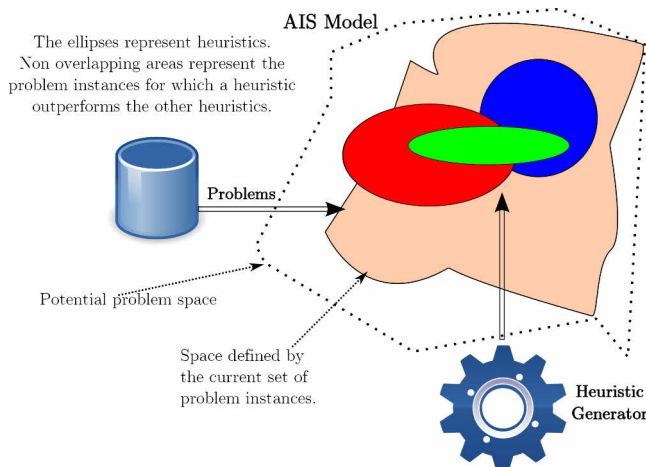


Figure 1: System Model

The system is designed to run continuously; problem instances can be added or removed from the system at any point. A heuristic generator akin to *gene libraries* in the natural immune system provides a continual source of potential heuristics. The AIS itself consists of a set of heuristics (akin to *antibodies in the natural immune system* that interact with each other based on an *affinity metric*. The overall goal of

the system is to develop a repertoire of heuristics that can solve the set of problem instances to which they are currently exposed and that can adapt its structure and constituent parts as the problem instances change in nature. The component parts are described in detail below.

## Heuristic Generator

Heuristics are represented using a tree structure and are generated using only the *initialisation* process used in SNGP (Jackson, 2012) resulting in structures as shown in the example given in Figure 2 — this tree shows one of the deterministic heuristics from the literature DJD encoded in a tree format. A fixed set of terminal and function nodes are available to the generator and are defined in Table 2 which combines nodes according to the process outlined in Algorithm 1. Further details outlining the justification for the choice of nodes and further details on the SNGP process can be found in Sim and Hart (2013). One heuristic is generated per iteration of the AIS algorithm.

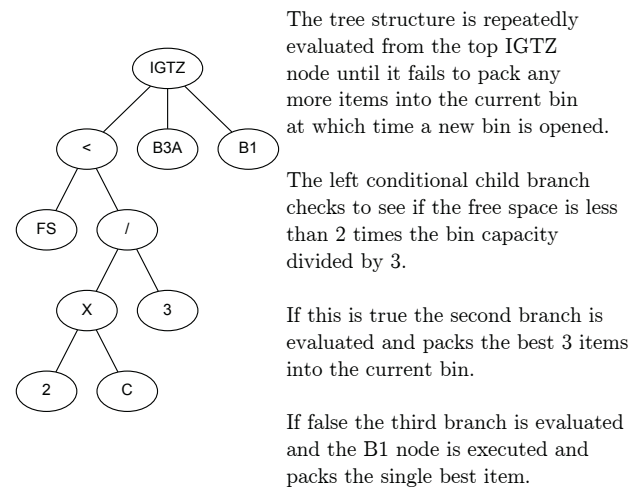


Figure 2: DJD Heuristic Expressed as a Tree

## Algorithm 1 Heuristic Generation

- 1: Each of the terminal nodes  $T \in \{t_1, \dots, t_r\}$  are added exactly once. The terminal nodes are given an integer identification number ranging from  $1 \dots r$ .
- 2: A number,  $n$ , of function nodes are selected at random from the set of all function nodes  $F \in \{f_1, \dots, f_s\}$  and given an identification number ranging from  $r + 1, \dots$  to  $r + n$ . This allows for the possibility of duplicate function nodes within the population or for SNGP structures with function nodes omitted.
- 3: The function nodes have all their child nodes assigned at random from nodes with a lower id thus preventing any infinite looping.
- 4: A single node is chosen at random to be the root node.



Table 1: Data sets *ds1*, *ds3* and *FalU* were created by generating  $n$  items with weights randomly sampled from a uniform distribution between the bounds given by  $\omega$ . Those in *FalT* were generated in a way similar to Falkenauer (1996) so that the optimal solution has exactly 3 items in each bin with no free space. Scholl's *ds2* was created by randomly generating weights from a uniform distribution in the range given by  $\varpi \pm \delta$ . The final column gives the number of instances generated for each parameter combination.

| Data Set    | capacity ( $c$ ) | $n$              | $\omega$                  | #Problems            |
|-------------|------------------|------------------|---------------------------|----------------------|
| <i>ds1</i>  | 100,120,150      | 50,100,200,500   | [1,100],[20,100],[30,100] | $36 \times 20 = 720$ |
| <i>ds3</i>  | 100000           | 200              | [20000,30000]             | 10                   |
| <i>FalU</i> | 150              | 120,250,500,1000 | [20,100]                  | $4 \times 20 = 80$   |
| <i>FalT</i> | 1                | 60,120,249,501   | [0.25,0.5]                | $4 \times 20 = 80$   |

| Data Set   | $c$  | $n$            | $\varpi$ (avg weight)                                | $\delta$ (%) | # Problems           |
|------------|------|----------------|------------------------------------------------------|--------------|----------------------|
| <i>ds2</i> | 1000 | 50,100,200,500 | $\frac{c}{3}, \frac{c}{5}, \frac{c}{7}, \frac{c}{9}$ | 20,50,90     | $48 \times 10 = 480$ |

Table 2: Nodes Used

#### Function Nodes

|      |                                                                                                                                                                                            |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /    | Protected divide returns -1 if denominator is 0 otherwise the result of dividing the first operand by the second                                                                           |
| >    | Returns 1 if the first operand is greater than the second or -1 otherwise                                                                                                                  |
| IGTZ | Evaluates the first operand. If it evaluates as greater than zero the result of evaluating the second operand is returned otherwise the result of evaluating the third operand is returned |
| <    | Returns 1 if the first operand is less than the second or -1 otherwise                                                                                                                     |
| X    | Returns the product of two operands                                                                                                                                                        |

#### Terminal Nodes

|     |                                                                                                                                                             |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| B1  | Packs the single largest item into the current bin returning 1 if successful or -1 otherwise                                                                |
| B2  | Packs the largest combination of exactly 2 items into the current bin returning 1 if successful or -1 otherwise                                             |
| B2A | Packs the largest combination of up to 2 items into the current bin giving preference to sets of lower cardinality. Returns 1 if successful or -1 otherwise |
| B3A | As for B2A but considers sets of up to 3 items                                                                                                              |
| B5A | As for B2A but considers sets of up to 5 items                                                                                                              |
| C   | Returns the bin capacity                                                                                                                                    |
| FS  | Returns the free space in the current bin                                                                                                                   |
| INT | returns a random integer value $\in (-1, \dots, +5)$                                                                                                        |
| W1  | Packs the smallest item into the current bin returning 1 if successful else -1                                                                              |

## AIS

The AIS component is responsible for constructing a network of interacting heuristics and for governing the dynamic

processes that enable heuristics to be incorporated or rejected from the current network. These two aspects are now described.

**Affinity** A key aspect of this is the *affinity metric* that defines the manner and the extent to which heuristics can interact. In the natural immune system, affinity is defined by physical and chemical interactions between molecules of different shape, with molecules with *complementary* structures showing highest affinity (the well known lock-and-key analogy). We re-interpret the notion of complementarity in the heuristic space as follows:

**Definition** Heuristic  $H_A$  is complementary to Heuristic  $H_B$  if Heuristic  $H_A$  uses fewer bins than Heuristic  $H_B$  on at least one problem instance

The *affinity* of Heuristic  $H_A$  for Heuristic  $H_B$  is equal to the number of extra bins  $\Delta b_{abp}$  used by Heuristic  $H_B$  summed across the set of  $l$  problem instances available to the system, and is defined in equation 2. Note that affinity between two heuristics is asymmetrical.

$$\alpha_{ab} = \sum_{p=1}^l \Delta b_{abp} \begin{cases} \Delta b_{a_p b_p} = b_{b_p} - b_{a_p} & : \text{if } b_{a_p} < b_{b_p} \\ \Delta b_{a_p b_p} = 0 & : \text{otherwise} \end{cases} \quad (2)$$

The total *stimulation* experienced by a heuristic is the sum of the affinities with all other heuristics in the network and is given by Equation 3 where there are  $m$  heuristics present in the network.

$$s_x = \sum_{j=1}^m \alpha_{xj} \quad (3)$$

An example for a system of  $m = 3$  heuristics and  $l = 3$  problems is given in figure 3.

**Network Dynamics** Each iteration, one new heuristic is generated and is made available to the network. The affinity metric described encourages *diversity* between pairs of

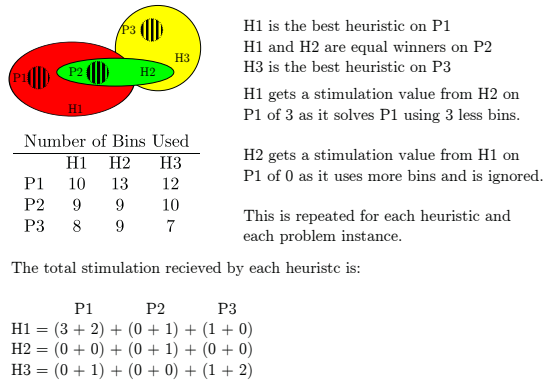


Figure 3: Affinity between Complementary Heuristics

heuristics leading to increased network performance by sustaining those that cover different parts of the problem space. In a practical application however, it is reasonable to assume that in addition to maintaining diversity, important goals of the system should be to find (1) the set of heuristics that most efficiently cover the problem space and (2) the set that collectively minimise the total number of bins used to solve all problems the network is exposed to. While the latter is addressed by sustaining any heuristic with non-zero stimulation, the former goal requires some attention.

Previous AIS models relating to idiotypic networks attempt to use Farmer's original equation (Farmer et al., 1986) to govern the dynamics of addition and removal of nodes from a network. In machine-learning applications such as data-clustering this was quickly found to lead to population explosion e.g. Timmis et al. (2000), later addressed by using resource limiting mechanisms (Timmis and Neal, 2001). In previous robotic applications, the situation is avoided completely by using a network of fixed size and focusing only on evolving connections. In more theoretical models such as Hart (2006) the criteria are not relevant, as the goal is simply to show that a network can be sustained. In this heuristic case, simply sustaining all heuristics that contribute to covering the heuristic space is likely to lead to population explosion in the same manner observed in data-mining applications, as no pressure exists on the system to encourage efficiency.

Therefore, at the end of each iteration, the contribution made by each heuristic to the overall performance is calculated in terms of whether it plays a unique contribution in determining the overall quality of the system. Any heuristic whose contribution is subsumed by one or more other heuristics is removed from the system. Thus, in Figure 3 heuristic  $H2$  is subsumed by the combination of  $H1$  and  $H3$  and is therefore removed from the system. This simple method thus provides pressure on the system to minimise the number of heuristics used. This can be thought of as an artificial form of apoptosis that removes antibodies that

cover duplicate parts of the landscape and allows for proliferation of new cells only in the case that they cover an equal or greater area of the search space that an existing heuristic is occupying.

Pseudo-code describing the network dynamics is given in Algorithm 2.  $\mathcal{H}$  is the set of heuristics currently present in the network,  $\mathcal{E}$  is the set of problems in the current environment. Note that there is a single parameter in the algorithm that defines the maximum concentration a heuristic can reach. This parameter introduces user control over the period of the network *memory* and is discussed in section .

---

**Algorithm 2** AIS Pseudo Code
 

---

```

1: repeat
2:   Add a randomly generated heuristic
3:   Optionally change the set of problem instances
4:   for all heuristics  $i \in \mathcal{H}$  do
5:     Calculate current  $stimulation_i$  using Equation 3
       based on  $\mathcal{E}$ 
6:     if  $stimulation_i > 0$  then
7:       if  $concentration_i < concentration_{max}$  then
8:          $concentration_i \leftarrow concentration_i + 1$ 
9:       end if
10:    else
11:       $concentration_i \leftarrow concentration_i - 1$ 
12:    end if
13:  end for
14:  Remove heuristics with  $concentration \leq 0$ 
15:  Remove all heuristics that give no global improvement
      (oldest first)
16: until stopping criteria met

```

---

## Experimental Results

A number of experiments were conducted using the model described with the test set of 1370 instances previously described in section .

### Baseline comparison on a static dataset

An initial experiment was performed in order to compare the AIS model to previous work in terms of solution quality on a *static* data set. 1370 instances were split into two sets: the *training* set comprised of the first and then every second problem instance from each of the data sets with the remaining problem instances used as a test set. This split ensured an even distribution of problem instances from each of parameter combinations used to generate them in the test and training set. All 685 problems in the training set were placed in the AIS environment, and the AIS algorithm run for 200 iterations. The quality of the network is measured in terms of the number of problem instances for which the known optimal solution was found by at least one heuristic, and additionally in terms of the number of extra bins

required to solve the instances compared to the known optimal. Results are recorded for problem instances in the training set, and then on the test set of instances (with no further iteration of the network). The experiments are repeated 30 times, reinitialising the system each time. Results are compared to using each of the single deterministic heuristics and to the island model described in Sim and Hart (2013) and shown in table 3a.

The best results shown for the AIS are identical to the best results presented in Sim and Hart (2013), even though the Island Model in that publication required evolutionary operators in order to find solutions and the AIS relies simply on randomly generated heuristics. Given this performance, we now examine the response of the system to dynamically changing data.

### Response to dynamically changing data

The AIS method was inspired by the ability of the natural immune system to rapidly respond to a dynamically changing environment, in which the self-sustaining network of antibodies is postulated to act as a memory of past responses. Four experiments were devised in order to investigate the behaviour of the system under the following dynamically changing conditions.

1. A set of 685 randomly selected problem instances is introduced every 200 iterations. At this point, problems currently in the system are removed, the network cleared, and the system is started from scratch
2. A set of 685 randomly selected problem instances is introduced every 200 iterations. At this point, problems currently in the system are removed, but the existing network is retained
3. Every 200 iterations the problem instances used are toggled between those from Scholl's data sets 1 & 2 (described in Table 1). Antibodies in the existing network are retained.
4. A new problem is introduced every iteration. The existing network is retained.

**Experiment 1** was designed to investigate the time taken for the system to reach equilibrium from an initial starting state. The results are shown in Figure 4. Note that in this graph (and in the following ones) only the first 5 cycles are shown. At the end of each 200 iteration cycle, the total number of bins required by the system is *assumed* to be the best that can be achieved and is given a score of 0. The y-axis then shows the number of bins required over and above this score, enabling different sets of problem instances where the optimal number of bins required varies to be plotted on a relative scale to highlight the response to changing conditions.

It is clear that the system performs poorly at the point of restart requiring up to 1200 extra bins than the best result

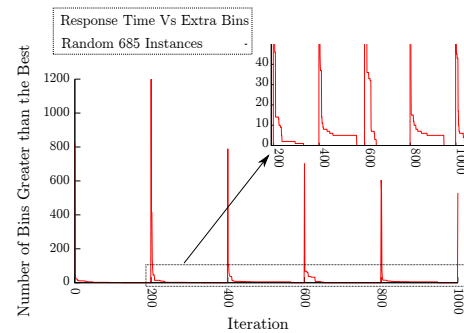


Figure 4: Response when the system is completely restarted every 200 iterations

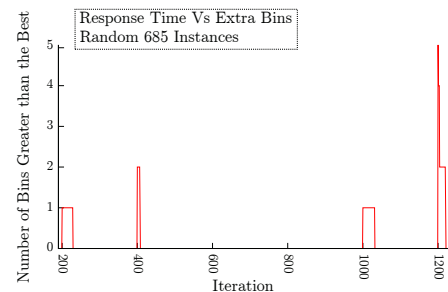


Figure 5: Response When the Problem Instances are Changed every 200 iterations

found during each cycle. The response is still rapid however: the median number of iterations required to reach the best result is 65 (shown in table4 ).

**Experiment 2** examines the role of memory, implicit in the sustained network. The system is effectively trained during a 200 iteration cycle on a set of problem instances representative of the whole set of 1370 problem instances. When the problem instances are changed, the heuristics already in the system have a greater probability of performing well on the new set of problem instances than randomly generated ones due to the shared problem characteristics. Results are shown in Figure 5. The median number of iterations to reach the optimal value is 10 and the total number of bins required is at most 5 greater than that at the end point (Table 4).

**Experiment 3** presents a more difficult test for the system; the two alternating problem sets have very different characteristics. The memory encapsulated in the network is therefore expected to be of less relevance. Results are shown in Figure 6 and in Table 4. Performance drops in comparison to Figure 5 where the memory could be exploited; however, there is still an improvement in comparison to the complete re-start applied in Experiment 1.

Finally, **Experiment 4** investigates the system response to slowly changing environmental conditions. Figure 7 shows how the system responds as problem instances are gradually introduced to the system. The system starts with no problem instances and no heuristics. Each iteration one randomly

| Single Deterministic Heuristics |                 |            |                     | Collaborative Heuristic Models |            |      |     |            |     |      |     |
|---------------------------------|-----------------|------------|---------------------|--------------------------------|------------|------|-----|------------|-----|------|-----|
| Heuristic                       | Problems Solved | Extra Bins |                     | Problems Solved                |            |      |     | Extra Bins |     |      |     |
|                                 |                 |            |                     | min                            | max        | mean | sd  | min        | max | mean | sd  |
| FFD                             | 393             | 1088       | <b>Immune Model</b> | 554                            | <b>559</b> | 556  | 1.4 | <b>159</b> | 165 | 162  | 1.4 |
| DJD                             | 356             | 1216       |                     |                                |            |      |     |            |     |      |     |
| DJT                             | <b>430</b>      | <b>451</b> | <b>Island Model</b> | 552                            | <b>559</b> | 557  | 1.4 | <b>159</b> | 164 | 162  | 1.4 |
| ADJD                            | 336             | 679        |                     |                                |            |      |     |            |     |      |     |
| BFD                             | 394             | 1087       |                     |                                |            |      |     |            |     |      |     |
| SS                              | 383             | 1112       |                     |                                |            |      |     |            |     |      |     |

(a)

(b)

(a)

(b)

Table 3: A comparison of results obtained on a static dataset of 685 problems using a) single heuristics and b) collaborative methods

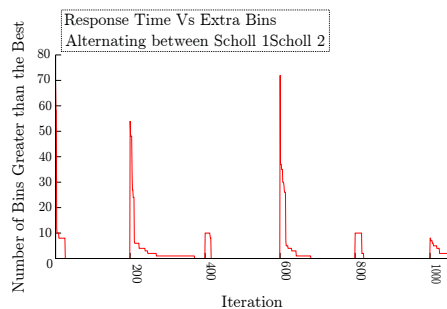


Figure 6: Response When alternating between data sets every 200 iterations

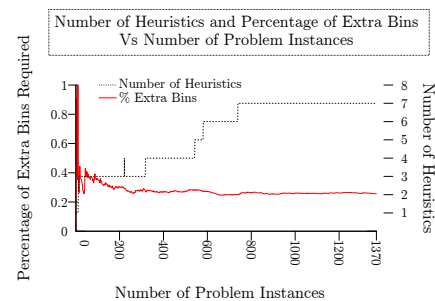


Figure 7: Number of Heuristics Sustained and the Percentage of Extra Bins Required than Optimal as the Number of Problem Instances is Increased

Table 4: System Response Time to Varying Conditions

|             | Response Time |     |        |
|-------------|---------------|-----|--------|
|             | min           | max | median |
| <b>Exp1</b> | 9             | 186 | 63     |
| <b>Exp2</b> | 1             | 111 | 10     |
| <b>Exp3</b> | 15            | 200 | 64.5   |

Exp1, 2 & 3 correspond to Figures 4, 5 & 6 respectively. The minimum, maximum, and median number of iterations that were required to reach the optimal value are shown. All results are taken over 30 data points.

generated heuristic and one randomly selected problem instance are introduced. The graph shows the percentage of extra bins over the known optimal that are required using the heuristics present in the network at each iteration. The number of heuristics sustained by the system is also shown.

Most benefit is gained from adding new heuristics when few heuristics are present. As the number of heuristics increases, it becomes harder for a newly added heuristic to

find a new niche. Furthermore, the total number of potential heuristics is limited by the generation method currently used (although this could easily be extended by adding new nodes).

## Conclusion

We have introduced an AIS model for generating and maintaining a repertoire of heuristics that solve bin-packing problems. Results show that the model achieves equal results to state-of-the-art methods on static data sets, and is extremely responsive to dynamically changing data-sets, making it suitable for use as a continuous learning system. The memory encapsulated in the network can be exploited to provide a rapid response to new problems that share characteristics with those previously seen by the network.

The model can be generalised to other domains by replacing the component parts of the heuristic generator. Currently, the generation model does not make use of any evolutionary or cloning operators to improve randomly generated solutions. By adding this feature in the future, it is hoped to speed up the response even further.



## Acknowledgements

This work was funded by EPSRC grant P/J1021628/1 Real World Optimisation with Life-Long Learning.

## References

- Burke, E., Hyde, M., and Kendall, G. (2006). Evolving bin packing heuristics with genetic programming. In *Parallel Problem Solving from Nature - PPSN IX*, volume 4193 of *Lecture Notes in Computer Science*, pages 860–869. Springer Berlin / Heidelberg.
- Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P., and Schulenburg, S. (2003). Hyper-heuristics: An emerging direction in modern search technology. In *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, chapter 16, pages 457–474. Kluwer.
- Burke, E. K., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., and Woodward, J. R. (2010). A classification of hyper-heuristic approaches. In Gendreau, M. and Potvin, J.-Y., editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, pages 449–468. Springer US.
- Burke, E. K., Hyde, M. R., Kendall, G., and Woodward, J. (2012). Automating the packing heuristic design process with genetic programming. *Evol. Comput.*, 20(1):63–89.
- Burnet, F. M. (1959). *The clonal selection theory of acquired immunity*. Cambridge University Press, Cambridge, UK.
- Falkenauer, E. (1996). A hybrid grouping genetic algorithm for bin packing. *Journal of Heuristics*, 2:5–30.
- Farmer, J. D., Packard, N. H., and Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Phys. D*, 2(1-3):187–204.
- Gent, I. P. (1998). Heuristic solution of open bin packing problems. *Journal of Heuristics*, 3(4):299–304.
- Hart, E. (2006). Analysis of a growth model for idiotypic networks. In Bersini, H. and Carneiro, J., editors, *Artificial Immune Systems*, volume 4163 of *Lecture Notes in Computer Science*, pages 66–80. Springer Berlin / Heidelberg.
- Jackson, D. (2012). Single node genetic programming on problems with side effects. In Coello, C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., and Pavone, M., editors, *Parallel Problem Solving from Nature - PPSN XII*, volume 7491 of *Lecture Notes in Computer Science*, pages 327–336. Springer Berlin Heidelberg.
- Jerne, N. K. (1974). Towards a network theory of the immune system. *Ann Immunol (Paris)*, 125C(1-2):373–89.
- Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA.
- Kromer, P., Platos, J., and Snasel, V. (2012). Practical results of artificial immune systems for combinatorial optimization problems. In *Nature and Biologically Inspired Computing (NaBIC), 2012 Fourth World Congress on*, pages 194–199.
- Neal, M. (2003). Meta-stable memory in an artificial immune network. In Timmis, J., Bentley, P., and Hart, E., editors, *Artificial Immune Systems*, volume 2787 of *Lecture Notes in Computer Science*, pages 168–180. Springer Berlin Heidelberg.
- Potter, M. A. and De Jong, K. A. (2000). Cooperative co-evolution: An architecture for evolving coadapted sub-components. *Evol. Comput.*, 8:1–29.
- Scholl, A., Klein, R., and Jürgens, C. (1997). Bison: a fast hybrid procedure for exactly solving the one-dimensional bin packing problem. *Comput. Oper. Res.*, 24(7):627–645.
- Schwerin, P. and Wäscher, G. (1997). The bin-packing problem: A problem generator and some numerical experiments with ffd packing and mtp. *International Transactions in Operational Research*, 4(5-6):377–389.
- Sim, K. and Hart, E. (2013). Generating single and multiple cooperative heuristics for the one dimensional bin packing problem using a single node genetic programming island model. In *Proceedings of GECCO 2013*, New York, NY, USA (to appear). ACM.
- Timmis, J. and Neal, M. (2001). A resource limited artificial immune system for data analysis. *Knowledge-Based Systems*, 14(34):121 – 130.
- Timmis, J., Neal, M., and Hunt, J. (2000). An artificial immune system for data analysis. *Biosystems*, 55(13):143 – 150.
- Watanabe, Y., Ishiguro, A., and Uchikawa, Y. (1998). Decentralized behavior arbitration mechanism for autonomous mobile robot using immune network. In Das-Gupta, D., editor, *Artificial Immune Systems and Their Applications*, pages 187 – 209. Springer-Verlag New York, Inc.
- Whitbrook, A. M., Aickelin, U., and Garibaldi, J. M. (2010). Two-timescale learning using idiotypic behaviour mediation for a navigating mobile robot. *Appl. Soft Comput.*, 10(3):876–887.

# An immune network approach for self-adaptive ensembles of autonomic components: a case study in swarm robotics

Nicola Capodieci<sup>1</sup>, Emma Hart<sup>2</sup> and Giacomo Cabri<sup>1</sup>

<sup>1</sup>University of Modena and Reggio Emilia

<sup>2</sup>Institute for Informatics and Digital Innovation, Edinburgh Napier University

<sup>1</sup>{nicola.capodieci, giacomo.cabri}@unimore.it

<sup>2</sup>e.hart@napier.ac.uk

## Abstract

We describe an immune inspired approach to achieve self-expression within an ensemble, i.e. enabling an ensemble of autonomic components to dynamically change their coordination pattern during the runtime execution of a given task. Building on previous work using idiotypic networks, we consider robotic swarms in which each robot has a lymph node containing a set of antibodies describing conditions under which different coordination patterns can be applied. Antibodies are shared between robots that come into communication range facilitating collaboration. Tests in simulation in robotic arenas of varying complexity show that the swarm is able to learn suitable patterns and effectively achieve a foraging task, particularly in arenas of high complexity.

## Introduction

Current and emerging ICT scenarios increasingly rely on complex distributed software systems in order to function properly in dynamic and unpredictable environments Zambonelli et al. (2011). This results in a need for the software controlling such systems to become *autonomic* in adapting behaviours such that quality of service of the system is maintained.

In Zambonelli et al. (2011), the authors describe two important dimensions of adaptation that can occur within autonomic systems, which they refer to as *where* and *what*. *Where* relates to where adaptation takes place, i.e. at the individual or ensemble level. *What* on the other hand refers to the set of mechanisms the system can utilise to adapt. They distinguish between *self-adaptation* and *self-expression*: the former refers to components or ensembles modifying their parameters so as to exploit their current abilities, whereas the latter describes the ability of radically modifying at run-time the structure of components and ensembles. In terms of ensembles, self-expression could result in re-structuring in terms of topology (e.g., switching from a hierarchy to a collective of peers) or of control regime for interactions (e.g., switching from being a collective decision-making ensemble to a competitive market-based one) Zambonelli et al. (2011).

While self-adaptation mechanisms at both individual and ensemble levels has been the focus of much research (e.g.

see Salehie and Tahvildari (2009) for overviews), mechanisms for achieving self-expression in distributed systems are less well-understood, particularly with respect to systems that enable software within an ensemble to express at run-time the most useful interaction topology or control regime. Inspired by the fact the features and properties apparent in the natural immune system such as scalability, adaptivity through learning and decentralization map naturally to those desired in autonomic systems Cabri and Capodieci (2013), we describe an idiotypic-network approach to self-expression. We consider a swarm-robotic scenario in which multiple robots have to fulfil a simple foraging task. Multiple *coordination patterns*, i.e. collaborative strategies, are available to the robots to face the problem; robots contains a lymph node describing a set of antibodies that indicate a suitable strategy — by sharing of antibodies across the swarm when robots come into contact with each other, the entire swarm is able to learn to solve the problem over time, even when placed in environments of varying complexity.

## Previous Work

A fruitful line of work within robotics that started with Ishiguro et al. (1995) has applied inspiration from Jerne's idiotypic network theory to develop behaviour arbitration mechanisms in individual robots. Antibodies consist of a matching condition to match environmental conditions, an action, and of receptors that enable interactions with other antibodies. The resulting network of stimulatory and suppressive connections alters concentrations of antibodies; the one with the highest concentration applies its action. Various weaknesses in this work that required hand-coding of antibodies for instance have recently been addressed in Whitbrook et al. (2010b), who consider evolutionary methods for generating antibodies and reinforcement learning to connecting them in a network, resulting in a system that has been ported successfully to real-robots Whitbrook et al. (2010a). We extend this work in that we deal with *swarms* of robots rather than individuals, and that rather than considering individual actions, the robots must select a cooperative strategy to take



Figure 1: A footbot. Sensors and actuators are placed as follows: (1) omni-directional sensing camera, (2) proximity sensors, (3) LEDs, (4,5) wheels and ground sensors, (6) RAB sensor and actuator

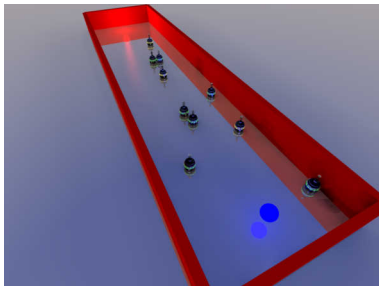


Figure 2: Simple example arena. Footbots in action are shown, together with the blue light symbolizing the nest and, in the opposite end, a red light for indicating the food area.

part in.

There have also been previous attempts to apply idiotypic network ideas to swarms. In Jun et al. (1999), a distributed version of Ishiguro et al. (1995) is proposed, in which individual robots choose an appropriate action according to a changing environment and the experiences of nearby robots. In Luh et al. (2006), instead of a single action, more complex behaviours are selected as a result of modelling a two layered immune network, merged as interaction among antibodies in the single robot and then distributed throughout all the swarm. Our work differs from both these publications, firstly in the robotic scenarios used and for switching *coordination patterns* as a response of the designed artificial immune network; this implies not only selecting behaviours, but also roles, statuses and interactions within the coordination pattern, and can result in dynamically changing interacting topologies, e.g. switching from a peer-to-peer coordination pattern to a purely stigmergic communicative collaborative effort and vice-versa.

| Sensor                                      | Purpose                                                                                                 |
|---------------------------------------------|---------------------------------------------------------------------------------------------------------|
| LED actuator                                | 12 light emitting diodes surround the robot. LEDs can be different colors                               |
| Omni-directional light sensing camera       | senses colored lights: returns their distance and angle of perception with respect to the sensing robot |
| Wheels actuator                             | enables movements                                                                                       |
| 24 proximity sensors                        | for detecting collisions                                                                                |
| Range and Bearing sensor and actuator (RAB) | sends/receives infra-red packets of a fixed 10 bytes size                                               |
| Ground sensor                               | detect strong variations in the floor color                                                             |

Table 1: Sensors and actuators modelled in robot simulation

## Task and robot description

We model a simple task in which a swarm of robots, initially randomly distributed in a confined arena, are required to collect food from a source and return that food to a nest, in an iterative process. The goal of the task is to maximise the total amount of food returned to the nest in a fixed time period, although performance can also be evaluated in terms of the rate of food collection over shorter time periods. The robot used in the simulation is called a *footbot*, and is one of the types of robot used in the swarmanoid project Dorigo et al. (2012) and in other previous research regarding swarm robotics coordination, e.g. Capodiceci and Cabri (2013) — see Figure 1. The sensors and actuators used on the robot are relatively simple. For instance, each robot is not able to calculate its position in space, nor does it have any concept of orientation. The modelled sensors and actuators are shown in table 1.

## Coordination Patterns

Three coordination patterns are considered: a completely swarm approach, a peer-to-peer (p2p) directly communicative approach and a baseline coordination strategy operated with the limited amount of sensors and actuators. The artificial immune system (AIS) enables the robots to select between these patterns based on their assessment of the complexity of their current environment and expected utility.

## RACO: Robotic Ant Colony Optimization

*RACO* is a robotic application of the well-known ant colony optimization family of algorithms Dorigo et al. (1996). In a typical ACO implementation a virtual pheromone trail is used to guide agents towards an object of interest; in a robotic scenario, as it is difficult for a robot to actively modify the environment by laying pheromones, some of the robots in the swarm adopt the role of pheromone: each robot

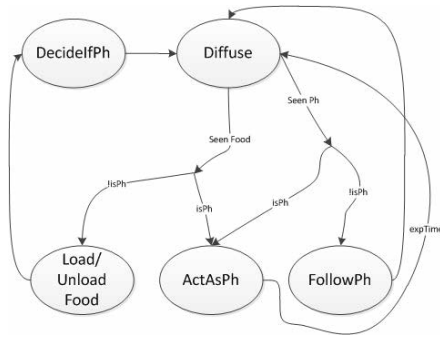
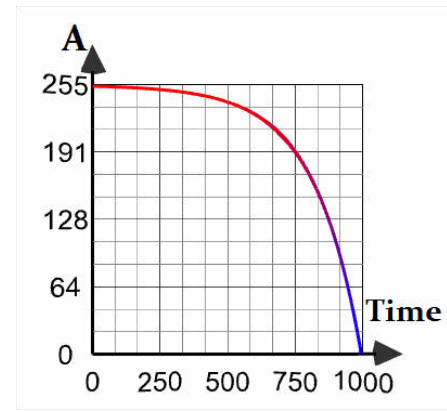


Figure 3: States and conditions for the RACO algorithm

is assigned a (small) probability  $p$  of acting a pheromone, with the remaining  $(1 - p)$  act as ants. All the robots start by uniformly diffusing into the area following a very simple diffusive algorithm: each robot starts with a random orientation and starts moving in a straight line; if a collision is sensed in one or more of the 24 proximity sensors the robot reacts by moving in the opposite direction from the sensed collision. If a robot that has decided to act as a *pheromone* reaches the food area (represented by a red emitting light) or the nest area (blue light), the robot stops moving and lights all 12 of its LEDs in yellow. Robots from the *ants* group continue to diffuse uniformly until they sense a pheromone; at this point they follow the pheromone trail rather than diffuse uniformly. The *intensity* of the pheromone is represented by the brightness of the emitted light. The more intense the light, the greater the distance from which it can be perceived by other robots. Intensity decays according to equation 1 which models evaporation over time (also shown in Figure 4).

$$Intensity(t) = MaxIntensity - \left( \frac{\beta e^{\frac{6t}{expTime}}}{1 - (\alpha - \beta)} + e^{(\alpha - \beta)(\frac{6t}{expTime})} \right) \quad (1)$$

In eq. 1,  $MaxIntensity$  represents the maximum value for the light intensity emitted by the robot's LEDs.  $expTime$  is the maximum expiration time (when the intensity reaches zero),  $t$  is the time variable and  $\alpha$  and  $\beta$  are constants. When the expiration time of the pheromone reaches zero, the *pheromone* robot switches its role to become an *ant* robot once more. However, if an *ant* robot comes within a distance  $d$  of the pheromone, the pheromone regenerates and its value is reset to  $MaxIntensity$ . When the expiration time of the pheromone reaches zero, the pheromone robot switches its role to become an ant robot (see Figure 3). Every robot makes its decision to act as a pheromone every time the food or nest area is reached, and that's how new pheromones are created.

Figure 4: Exponential evaporation time for the pheromone robot: emitted light intensity  $A$  over time

### AMORPH: Amorphous computing inspired path formation

The second coordination pattern strongly relies on communications in the form of packets sent and received with the RAB sensor and actuator built in each robot. This is purely peer-to-peer (p2p) approach in which roles are assigned dynamically according to the current situation. This strategy is inspired by the work of *Abelson et al.* in Abelson et al. (2000) which studies the use of bio-inspired algorithms for achieving collaboration among a potentially large number of devices connected in unknown way. The algorithm is described in detail in Abelson et al. (2000); the modifications required to adapt this to a robotic swarm are discussed below. Its underlying concept is to diffuse a gradient across the amorphous net of communicating robots indicating the shortest path from the nest to the food source.

The swarm starts by contracting who will be the path opener: any robot sensing the nest area (blue light) sends a message to nearby robots indicating its distance from the light. Any robot receiving a distance greater than the one it currently holds begins to diffuse; the process results in the single robot that is furthest from the light becoming the path-opener and remains stationary (robot A in figure 5). Robots diffuse uniformly until one robot reaches the nest area (red light) at which point it becomes the *path-closer* (robot-F in figure 5) and halts. This robot sends out a packet to nearby robots via infra-red signalling through the RAB actuator) consisting of the tuple (*SenderID*, *gradient*, *state*, *successorID*). *Gradient* is an indicator of the distance to the start of the path, *state* is a boolean value indicating whether a robot is currently part of the path, *successorID* is the ID of the robot that previously sent the highest gradient value to the receiving robot. The path-closer sends the maximum gradient value (255). Receiving robots store the highest value received  $g$ . They then transmit a new gradient  $g - 1$ . The chain ends when the *path-opener* receives a gradient and transmits



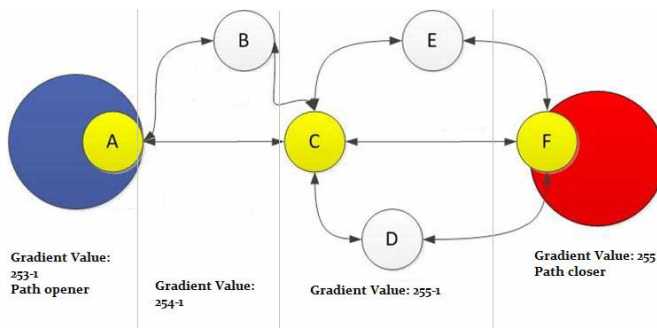


Figure 5: Schematic of an example situation in the amorphous path formation algorithm: communication is variable due to random placement, obstacles and limited RAB range for signalling. A possible communication topology is shown by the lines and arrows. At the end of the algorithm, robots A,C,F provide the path from food to nest area.

a message that a path is ready to be formed (setting the *state* packet to 1). All robots (including the *path-opener*) receiving state 1 set their LEDS to a specific colour thus indicating the path. Robots not on the path now follow the lit path to the food.

An emergent property of this algorithm is that the path of lit robots formed is exactly one-robot thick. The robots that are inside this path will be referred as *nodes*, while the other robots can still be called ant robots since their behaviour now is very similar to the one described in the previous coordination pattern.

### Baseline strategy: blind diffusion

An additional cooperative strategy is represented by a coordination pattern that uses the minimum amount of sensors and actuators. It is called blind diffusion, since all the robots are *blind*, in the sense that they disable the omni-directional camera, thus are unable to sense lights and colour for being attracted towards the point of interests in the arena.

All robots apply a simple diffusion algorithm: when a collision is sensed through the proximity sensor, they steer in opposite direction by rotating in a direction opposite to the angle to incidence w.r.t. their body centre of the proximity sensor detecting the collision. An emergent property of this algorithm is that the swarm becomes uniformly distributed throughout the whole arena. In this algorithm, food is sensed by ground sensors built in the wheels of each robot that detect floor colours based on a grey-scale. However, it should be clear that this algorithm will be much less effective than the previous approaches in terms of the amount of food collected.

### Remarks on coordination pattern performance

Previous experimentations with each of three coordination patterns in a small area led to a number of observations

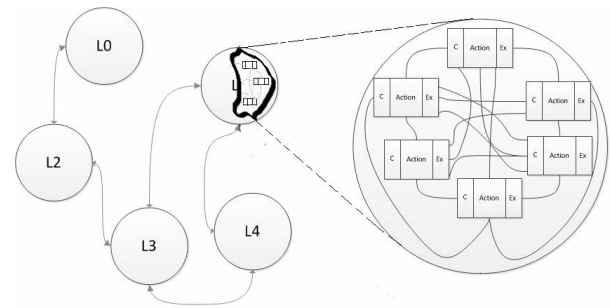


Figure 6: lymph-nodes are robots with variable connectivity and they can host a multitude of interconnected antibodies.

that motivates the selection strategy introduced in this paper. Briefly, it was observed that the AMORPH algorithm performs best (in terms of total food collected) in complex arenas, where complexity relates to both the size of the arena and the number of obstacles present. In simple (obstacle free) arenas, RACO performs best — probably due to the fact the the RACO algorithm enables more robots to be devoted to foraging, rather than being tied up in path construction. Both algorithms struggle in some situations as both lack a proper obstacle avoidance behaviour, therefore certain configurations of either node or pheromone robots can impede performance. This motivates the algorithm presented, which is not only able to dynamically alter the current coordination pattern due to online feedback regarding current performance, but also results in subsets of the swarm in fact following different coordination patterns at any given moment. This is described in the next section.

### SelfEx - Model

The model introduced extends work originally discussed in Ishiguro et al. (1995). Significantly, we extend the concept of controlling an individual robot through a network of antibodies that match conditions to actions to a distributed swarm in which the robots only have limited communication range. Each robot is modelled as a lymph node that contains a set of antibodies, connected in an idiotypic network, see Figure 6. Robots can diffuse antibodies from one lymph node to another, thus the contents of lymph nodes are continually adapted. In contrast to previous work, antibodies determine the *coordination pattern* that should be executed by the robot at a given moment rather than a specific action. The algorithm (**SelfEx**) is given in listing 1 and is described in detail below.

As we can see in fig. 6 each robot is modelled as a lymph-node in a net of lymph-nodes whose connectivity varies according to the ever changing position of the robots. By taking a look inside each robot, we can see how each lymph-node can host a multitude of interconnected antibodies and each antibody is characterized by a variable concentration value and it is divided into three main parts that will be ex-

plained in the following section.

---

**Algorithm 1** SelfEx: overview
 

---

- 1: **Initialisation**: each robot initialised with a set of antibodies
  - 2: **Maturation**: each robot estimates the complexity of the environment
  - 3: **Selection**: each robot selects a start coordination pattern based on its antibody that highest affinity with the environment
  - 4: **repeat**
  - 5:   **Evaluation & Affinity Update**: every *eval* timesteps, each robot evaluates its performance and updates the affinities between antibodies in its own idiotypic network
  - 6:   **Concentration Update**: each robot updates the concentration of each antibody within its lymph node
  - 7:   **Diffusion**: the best antibodies are diffused to robots within communication range
  - 8: **until** stopping criteria
- 

**Antibodies and Affinity** An antibody is a tuple of values  $\langle \text{Conditions}, \text{Action}, \text{expected utilities} \rangle$  as follows:

- **Condition** has two variables, *complexity* and *status*. *Complexity* is a real-value representing the complexity of the arena. *Status*  $\in \{0, 1, t\}$  where 0 indicates an ant robot in the RACO model, 1 a node robot in the AMORPH model and *t* indicates the time a robot has spent in the pheromone state in the case of pheromone robots.
- **Action**  $\in \langle \text{BlindDiffusion}, \text{RACO}, \text{AMORPH} \rangle$ .
- **Utility** is the value of the food collected in a time period, *eval* by each robot

The *affinity* between an antibody and the current environment is calculated as the Manhattan distance between the calculated *complexity* and the *complexity* value stored in the antibody.

**Initialisation** A set of antibodies to seed networks were derived from an initialisation phase; during this phase, a swarm utilised each individual coordination pattern in each tested arena over a 50000 time-step period. During this period, each robot logged its status, perceived complexity index and the amount of food it managed to collect during the evaluation time. These results were then averaged to build an initial set of 10 seed antibodies. Each robot's lymph node is loaded with the same set of 10 antibodies found. The initial concentration of each antibody in each lymph node is set to zero.

**Maturation** The purpose of the maturation phase is for each robot to estimate the complexity of the environment. This is calculated according to algorithm 2. Note that the lower the value of *complexity* calculated, the higher the complexity of the arena.

---

**Algorithm 2** SelfEx: Maturation Phase
 

---

- 1: initialise lymph node of each robot with a set of antibodies
  - 2: Goal  $\in \{Food, Nest\}$
  - 3: **repeat**
  - 4:   **for all** robot *i* in  $\mathcal{R}$  **do**
  - 5:     Move according to *blind diffusion pattern*
  - 6:     **if** Goal encountered **then**
  - 7:       *maturationCounter*<sub>*i*</sub>  $\leftarrow$  *maturationCounter* + 1
  - 8:     **end if**
  - 9:   **end for**
  - 10: **until** end of maturation phase
  - 11: *complexity* = averaged value of each robots *maturationCounter*
- 

**Evaluation and Affinity Update** After *eval* timesteps, each robot evaluates its own performance *u* and compares this to the expected utility *u<sub>E</sub>* indicated in the currently active antibody. If  $u > u_E$  then positive feedback is given to the selected antibody by increasing the affinities from all other antibodies towards the selected one; in addition to that, the expected utility field of this antibody is updated with the newly obtained value. If  $u < u_E$  then negative feedback results in the the selected antibody increasing its affinity to all other antibodies. Affinity values *r* (between antibody *i* and *j*) are calculated as follows:

$$r_{i,j} = \omega |obtUtilities - expUtilities| + \frac{K_0}{|dAc - abAc|}$$

$$\omega = \begin{cases} status & \text{if } status \leq 1 \\ 1 - \frac{status}{evalTime} & \text{if } status > 1 \end{cases} \quad (2)$$

The difference between obtained utilities (*obtUtilities*) and expected utilities (*expUtilities*) is weighted according to the *status* variables (see section ). In addition, the affinity is adjusted according to the difference in the detected area complexity and the complexity value stored in the antibody, regulated by the constant  $K_0$  ( $< 1$ ).

**Concentration Update** Antibody concentration updates are performed in a similar manner to that of Ishiguro et al. (1995) as shown in eq. 3. The main differences are that we assume that every antibody is connected to all the other antibodies, antibodies with low concentration are not removed, and we do not add new antibodies.

$$\frac{dc_i}{dt} = K_1 \sum_{j=0}^N r_{j,i} c_i c_j - K_2 \sum_{k=0}^N r_{i,k} c_i c_k + K_3 D c_i \quad (3)$$

The first term of eq. 3 refers to the stimulation part of the immune net; the second term (with a negative sign) refers to the suppression operated by the other antibodies and the latter term takes into account the euclidean distance  $D$  among all the conditions of the antibodies to the obtained/detected values of status, utilities and area complexity. Three constants ( $K_{1,2,3}$ ) regulates the contribution of each of these effects. The resulting concentration is then squashed to fit the  $[1...255]$  range.

**Diffusion** The calculated concentrations of the four antibodies with highest concentration in each lymph node are now shared amongst other robots in range, in order to distribute knowledge throughout the ensemble regarding current performance in a single packet<sup>1</sup>.

*Ant* robots *broadcast* such packets but do not *receive* packets as they are able to form their own estimate of how much food has been collected. *Pheromone* and *Node* robots average its concentration with the external concentration sent by other robots for each antibody. Following the updating step, the robot selects the antibody with the highest concentration and executes the coordination pattern indicated.

## Experiments and simulations

The pre-experimentation phase to establish the initial set of antibodies was undertaken in two areas; the first was characterized by the highest complexity index (visible in fig. 2, the simplest area), while the second arena with a high-index used an extensive sized hexagonal arena with an obstacle between the straight path from nest to food area. Tests with the main algorithm were performed in three different arenas: the two described above and one of intermediate complexity. Each experiment was repeated 10 times, with a different random initial distribution of robots each time. In each arena, experiments were performed with the single RACO and AMORPH algorithms as well as with SelfEx. Parameters were tuned through an empirical process — values used in the results reported are shown in table 2. The experiments were performed using *ARGoS* swarm robotics simulator (Pinciroli et al. (2012)).

## Results and remarks

As stated previously, the experiments were evaluated on the basis of total performances over the total length of the experiment and the variation of food collected each evaluation period (by the entire swarm). This latter metric is useful to note the gradual improvements of each collaborative effort:

<sup>1</sup>Only four antibodies are distributed due to the limitations of the RAB sensor

| Parameter | Description                         | Value   |
|-----------|-------------------------------------|---------|
| eval      | evaluation period for affinity step | 10,000  |
| l         | total timesteps experiment run      | 200,000 |
| m         | maturation phase length             | 5500    |
| $K_0$     | from equation 2                     | 0.05    |
| $K_1$     | from equation 3                     | 0.25    |
| $K_2$     | from equation 3                     | 1       |
| $K_3$     | from equation 3                     | 0.05    |
| $p$       | probability to become a pheromone   | 0.35    |
| $expTime$ | from equation 1 (RACO)              | 1500    |
| $\alpha$  | from equation 1 (RACO)              | 0.3     |
| $\beta$   | from equation 1 (RACO)              | 0.8     |

Table 2: Values of parameters used for the simulations

the AIS approach should show an average increasing trend in the amount of food collected in each time interval, in order to indicate the network is *learning*. Figure 7 shows the results of the simplest arena. SelfEx and RACO have similar performance, although SelfEx is less variable than RACO. AMORPH performs best in terms of the median value of food collected and variability. However it is clear from the right hand figure that the performance of AMORPH deteriorates over time, whereas SelfEx and RACO show steady improvement, suggesting that given further time they may at least equal the AMORPH performance. In the second arena (Figure 8) RACO performs best. SelfEx performs better than AMORPH, particularly in terms of variability. As in the first arena, the performance of SelfEx increases over time, a pattern not clear in RACO and AMORPH. Finally, in the third and most difficult arena, the best performance is achieved by SelfEx (Figure 9). In this arena, performance does not increase over time, suggesting the ensemble reached consensus on the coordination pattern to adopt very early in the experiment.

Figure 10 shows the percentage of robots in each evaluation period that follow the different coordination patterns in Arenas 1 and 3 (figures from a single run). In Arena 1, the swarm is divided in its choice of pattern — interestingly, some robots still select the blind diffusion strategy. In contrast, in Arena 3, robots eventually converge towards a single coordination pattern. The ability of SelfEx to enable heterogeneous behaviours might be particularly useful in more dynamic environments.

## Conclusions and Future Work

We have introduced a novel approach to managing a collaborative swarm of robots by taking inspiration both from autonomic computing and bio-inspired algorithms, and extending previous work that used idiotypic network models to control individual robots. The model was tested on a simple foraging task in which robots had to choose a coordination pattern from a set of three different possibilities. Each robot

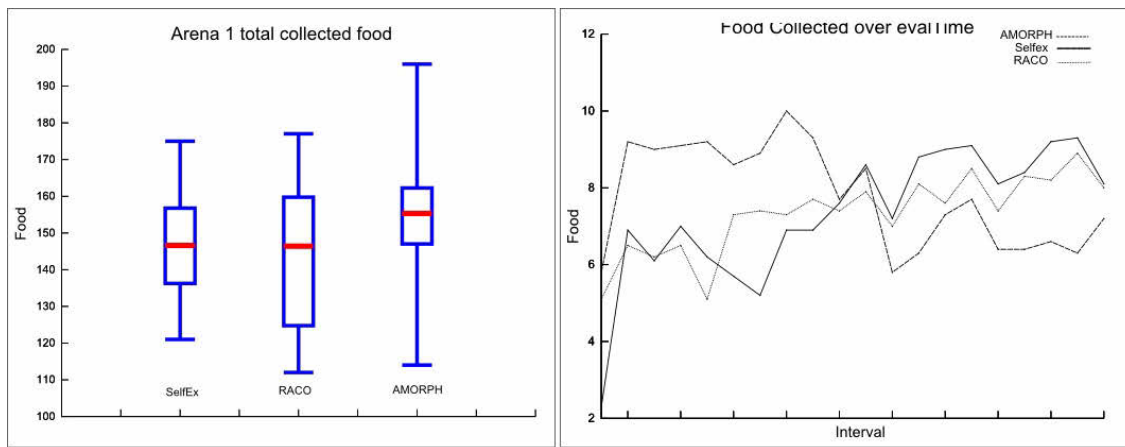


Figure 7: First test arena results

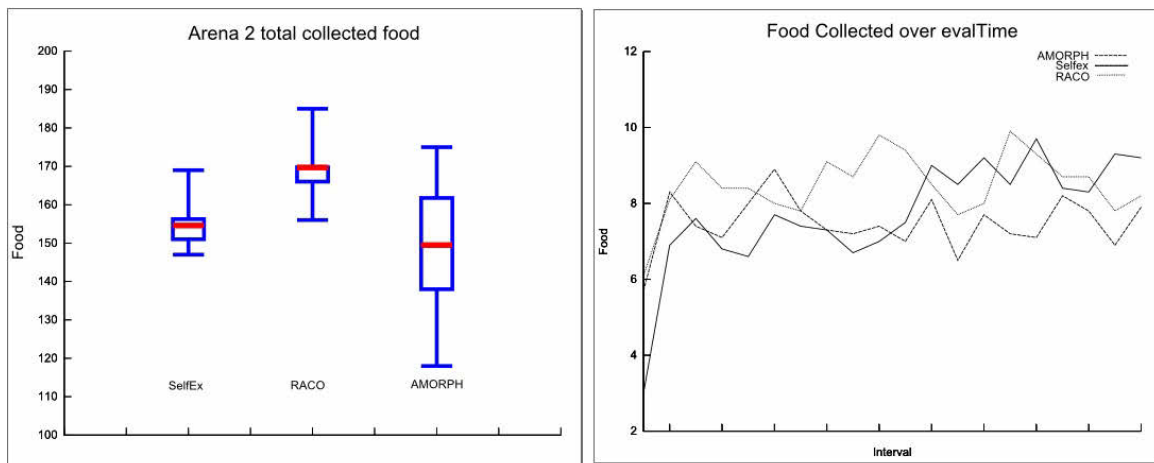


Figure 8: Second test arena results

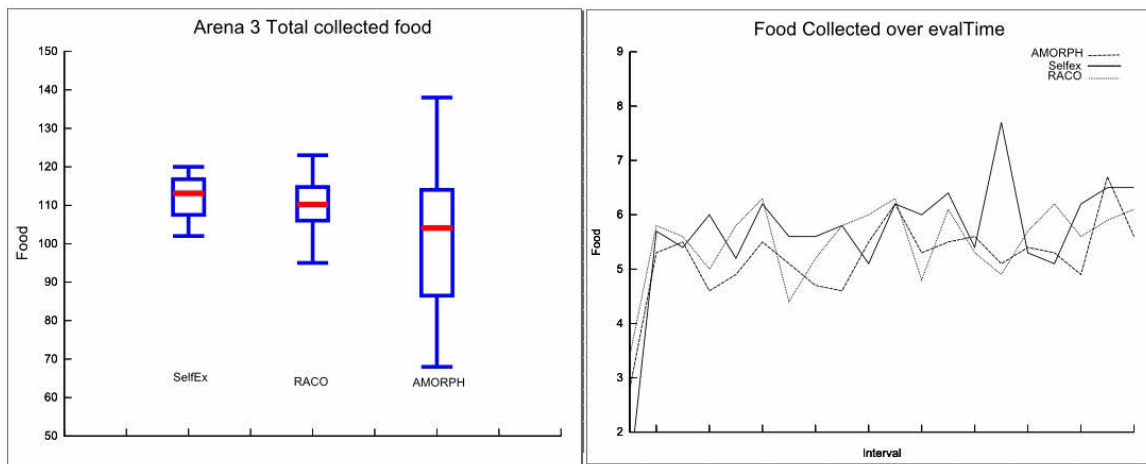


Figure 9: Third test arena results





Figure 10: Distribution of coordination patterns amongst robots in each time interval. The top figure shows Arena 1, the lower Arena 3. The figures are 'stacked' in the order blind diffusion, RACO, AMORPH from the bottom

is a modelled as lymph node hosting a connected network of antibodies. Sharing of antibodies across robots enables common decisions to be made as to the best manner to fulfill the task. The results shown that the ensemble was able to deploy Self-Expression, i.e. fragmenting into sub-sets and initially choosing different coordination pattern. The results show that the SelfEx approach was able to learn and improve its performance over time, thus demonstrating the cognitive ability as emerging property of an immune net. However, the overall performances also show that how our approach is slow to converge and the maturation phase implies smaller quantities of collected food during the initial time intervals: there is a large room for improvements by appropriately tuning the parameters involved. Additional improvement could be gained by using a hyper-mutation process in order to optimise individual antibodies within a network via a local search process.

**Acknowledgement** The work is partially supported by the ASCENS project (EU FP7-FET, Contract No. 257414)

## References

- Abelson, H. et al. (2000). Amorphous computing. *Communications of the ACM*, 43(5):74–82.
- Cabri, G. and Capodieci, N. (2013). Runtime change of collaboration patterns in autonomic systems. In *In Proc. of IEEE international Conference on Advanced Information Networking and Applications (AINA-2013) FINA 2013 workshop*. IEEE.
- Capodieci, N. and Cabri, G. (2013). Collaboration in swarm robotics: a visual communication approach. In *in proc. of The 2013 Conf. on Collaboration Technologies and Systems (CTS 2013), The 4th International Collaborative Robots and Human Robot Interaction workshop*, page to appear. IEEE.
- Dorigo, M. et al. (2012). Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics and Automation Mag.*
- Dorigo, M., Maniezzo, V., and Colomi, A. (1996). Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(1):29–41.
- Ishiguro, A., Watanabe, Y., and Uchikawa, Y. (1995). An immunological approach to dynamic behavior control for autonomous mobile robots. In *IEEE-RSJ International Conference on Intelligent Robots and Systems 95. Human Robot Interaction and Cooperative Robots*, volume 1, pages 495–500. IEEE.
- Jun, J., Lee, D., and Sim, D. (1999). Realization of cooperative strategies and swarm behavior in distributed autonomous robotic systems using artificial immune system. In *in Proc. of the IEEE International Conference on Systems, Man, and Cybernetics. IEEE SMC '99*, volume 6, pages 614–619. IEEE.
- Luh, G.-C., Wu, C.-Y., and Liu, W.-W. (2006). Artificial immune system based cooperative strategies for robot soccer competition. In *Strategic Technology, The 1st International Forum on*, pages 76–79. IEEE.
- Pinciroli, C. et al. (2012). Argos: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence, Springer Berlin*, 6(4):271–295.
- Salehie, M. and Tahvildari, L. (2009). Self-adaptive software: Landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 4(2):14.
- Whitbrook, A., Aickelin, U., and Garibaldi, J. (2010a). Real-world transfer of evolved artificial immune system behaviours between small and large scale robotic platform. *Evolutionary Intelligence*, 3(3):123–136.
- Whitbrook, A., Aickelin, U., and Garibaldi, J. (2010b). Two-timescale learning using idiotypic behaviour mediation for a navigating mobile robot. *Appl. SoftComput.*, 10(3):876–887.
- Zambonelli, F., Bicocchi, N., Cabri, G., Leonardi, L., and Puviani, M. (2011). On self-adaptation, self-expression, and self-awareness, in autonomic service component ensembles. in *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2011 Fifth IEEE Conference on*, Ann Arbor, Michigan, USA: IEEE Computer Society:108–113.

# An Artificial Immune System-based Many-Objective Optimization Algorithm with Network Activation Scheme

Wilburn W. P. Tsang<sup>1</sup> and Henry Y. K. Lau<sup>2</sup>

Department of Industrial and Manufacturing Systems Engineering,  
The University of Hong Kong, Hong Kong  
<sup>1</sup>wilburn@graduate.hku.hk <sup>2</sup>hyklau@hku.hk

## Abstract

In the research of multi-objective optimization algorithm, evolutionary algorithms have considered to be very successful tools. Artificial Immune System (AIS)-based algorithms as one of the viable alternative have also be widely developed in this domain. Over the years, researchers of evolutionary algorithms have extended their interest to many-objective situations; however works in AIS-based algorithms is rather scattered. This paper extends an AIS-based optimization algorithm to solve such many-objective optimization problems. The idea of  $\varepsilon$ -dominance and the holistic model of the immune network theory have been adopted to enhance the exploitation ability aiming for a quick convergence.

## AIS-based Many-objective Optimization Algorithm

Evolutionary algorithms (MOEA) have shown to be the state-of-art tools for solving multi-objective optimization problems with 2 or 3 objectives. The application has been extended to many-objective situation with 4 or more objectives. With modification in the fitness assignment process and the evolution scheme, these algorithms achieve satisfactory results.

AIS-based algorithms are considered to be one of the viable alternatives in multi-objective optimization. A number of AIS-based multi-objective optimization algorithms have been proposed with promising results (Shang, 2012; Tsang and Lau, 2012). Majority of these studies focus on 2 to 3 objectives, with many-objective situations rarely being considered. The experiment in (Jarosz and Burczynski, 2011) had extended the application to 5 objectives. However, the study of many-objective optimization with AIS is still rare. This study therefore attempts to develop a novel AIS-based algorithm for solving many-objective problems.

In general, an AIS-based multi-objective optimization algorithm first generates the initial population. This population will go through cloning, variation, evaluation, network suppression and memory updating. Cloning and variation process generates modified solutions. Evaluation process assesses the fitness of each solution. Different fitness assignment scheme was proposed to increase the chance of domination.  $\varepsilon$ -dominance is one of them which relaxes the dominance requirement (Hernández-Díaz, et al. 2007). The resulting solutions are often dominated by solutions with

lower or equal fitness in all objectives which are close enough to non-dominated solutions. Memory updating process selects population for next generation.

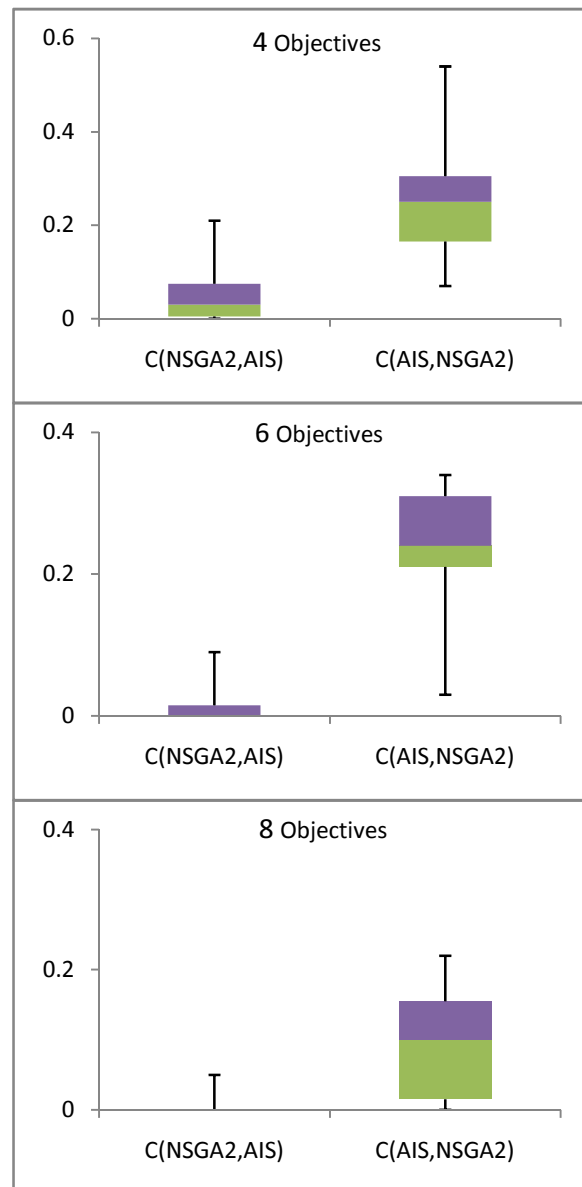
Inspired by the immune network theory, network suppression process provides a means to manage diversity. Such operation differentiates the AIS-based approaches from evolutionary algorithms in optimization. 'Near' solutions are suppressed to reduce redundant search. The concept of network interaction with suppression and activation had been implemented and tested on multi-objective optimization with promising results (Tsang and Lau, 2010). This study builds on the success in past studies and extends to many-objective situations. When a solution suppresses and dominates near solutions, this suppression hints a potential promising search if the exploitation can go further in the same direction from the dominated solution to the non-dominated solution. This suppression triggers the activation through generating new solutions on the identified high-potential space. The difference in decision variables define the search area and step size. New solutions will be generated based on the direction and step size as defined. Such an approach can supplement the dominance relation in identifying the searching area which accelerates the convergence through directed exploitation.

## Experiment and Conclusion

The AIS-based approach enhanced with the proposed network activation scheme has been implemented and compared against the general AIS-based approach and some other conventional approaches. Test problems from the DTLZ family (Deb, et al. 2002), which are scalable to different number of objectives, are used for studying the performance in continuous functions.

Preliminary results comparing with the NSGA-II (Deb, et al. 2000) and using the Coverage C metrics (Zitzler, 1999) are shown.  $C(A,B)$  gives the proportion of solutions in B that are dominated by solutions in A.  $C(\text{AIS}, \text{NSGA-II})$  is much higher than  $C(\text{NSGA-II}, \text{AIS})$  for all 3 cases of increasing number of objectives. Box-plot is used as the presentation tools as it displays also the summary of the whole result including the outliers. The results show the potential of the AIS-based many-objective optimization algorithm.

Figure 1 C metric between NSGA-II and AIS



## References

- Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J. and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature-PPSN VI*, pages 849-858. Springer
- Deb, K., Thiele, L., Laumanns, M. and Zitzler, E. (2002). Scalable multi-objective optimization test problems. In Yao, X., editor, *2002 Congress on Evolutionary Computation*, pages 825-830. IEEE
- Hernández-Díaz, A.G., Santana-Quintero, L.V., Coello Coello, C.A. and Luque, J.M. (2007). Pareto-adaptive epsilon-dominance. *Evolutionary Computation*, 15(4): 493-517
- Jarosz, P. and Burczynski, T. (2011). Artificial immune system based on clonal selection and game theory principles for multiobjective

optimization. In Lio, P., Nicosia, G. and Stibor, T., editors, *10th International Conference on Artificial Immune Systems (ICARIS 2011)*, pages 321-333. Springer

Shang, R., Jiao, L., Liu, F. and Ma, W. (2012). A novel immune clonal algorithm for MO problems. *IEEE Transactions on Evolutionary Computation*, 16(1): 35-50

Tsang, W.W.P. and Lau, H.Y.K. (2010). Enhanced network interaction in multi-objective immune optimization algorithm. In *8th International Conference on Optimization: Techniques and Applications*, Shanghai, China. December 10-13, 2010

Tsang, W.W.P. and Lau, H.Y.K. (2012). Clustering-based multi-objective immune optimization evolutionary algorithm. In Coello-Coello, C.A., Greensmith, J., Krasnogor, N., Liò, P., Nicosia, G. and Pavone, M., editors, *11th International Conference on Artificial Immune Systems (ICARIS 2012)*, pages 72-85. Springer

Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: methods and applications*. PhD thesis, Swiss Federal Institute of Technology, Zurich

## Automated calibration of agent-based immunological simulations

Mark Read<sup>1,\*</sup>, Magnus Tripp<sup>2</sup>, Hannah Leonova<sup>3</sup>, Louis M. Rose<sup>2</sup> and Jon Timmis<sup>1,2</sup>

<sup>1</sup>Department of Electronics, the University of York, UK.

<sup>2</sup>Department of Computer Science, the University of York, UK.

<sup>3</sup>Department of Chemical and Biological Engineering, the University of Sheffield, UK.

HL is now at the University of Sheffield; her contribution to this work was conducted at York.

\*mark.read@york.ac.uk

The emerging field of computational immunology shows great promise to advance immunological research. Simulations of immunological systems provide a platform for *in silico* experimentation, facilitating the formulation and evaluation of hypotheses.

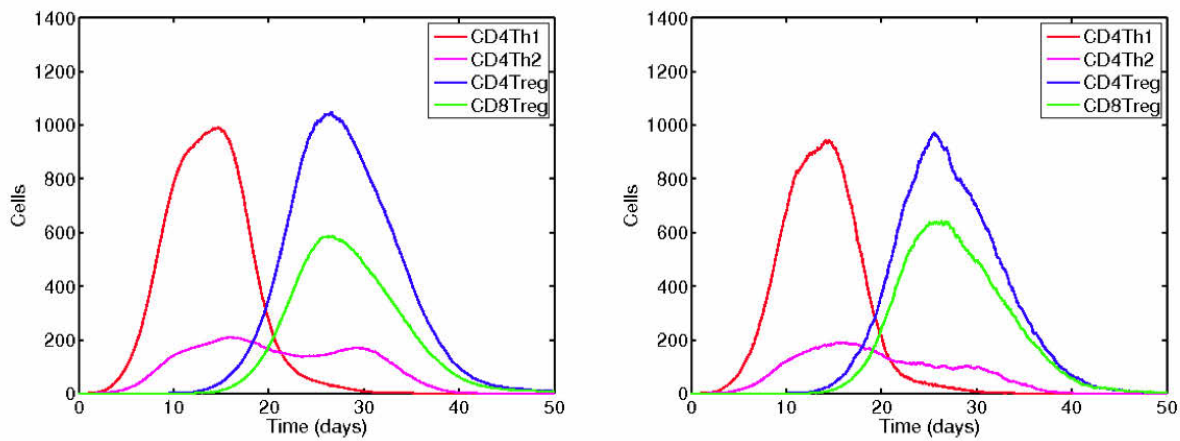
A major challenge in this field, however, lies in parameterization, particularly in agent-based simulations. These simulations contain many parameters (>50 is plausible), many pertaining to aspects of immunology that either have not or cannot be examined with current wet-lab technologies. Curve-fitting (e.g. linear regression) based-calibration is tractable only for relatively simple simulations with few parameters, and will not necessarily lead to biologically-plausible parameter values (e.g., if the model is a bad representation of the biology). For larger systems the current state of the art is to calibrate by hand/eye, with some values based on wet-lab data or expert opinion, and the rest on trial and error. Furthermore, it is typical to calibrate simulations against data from only a single wet-lab experiment. Although these data may comprise observations of multiple cells/molecules/disease scores (termed *responses*), given that a simulation is likely to be used to perform *multiple* novel experiments that have not been attempted in the wet-lab, it still constitutes calibration against a single data-point (single experiment). Put another way, with so many degrees of freedom there may be multiple points in parameter space for which a simulation re-creates data from a single experiment; a simulation calibrated in such a manner will not necessarily be representative of the biology when used for a different experiment. We propose that to have genuine trust that they reliably capture the biology, immunological simulations should be calibrated against multiple wet-lab experiments.

Performing calibration of this magnitude by hand is intractable, and as such we are investigating alternatives based on automated multi-objective meta-heuristic search techniques. Each wet-lab experiment used in calibration is performed also in simulation. Each response from each experiment is treated as an individual objective that the search algorithm must align simulation behaviour with. The

search algorithm searches for parameter values that satisfy all these constraints. We are developing this methodology by calibrating ARTIMMUS, an existing simulation of the murine autoimmune disease Experimental Autoimmune Encephalomyelitis (Read, 2011), using NSGA-II (K Deb, 2002), a multi-objective search technique. ARTIMMUS hand-calibrated parameters have been shown to reflect the *in vivo* disease dynamics of EAE (Read, 2011; Mark Read, 2012). ARTIMMUS comprises around 70 parameters, representing a very large search space. In demonstrating proof of principle of this technique we first calibrate a restricted set of 8 key parameters (shown in figure 1c; all other parameters retain their previously calibrated values, see Read (2011)), gradually increasing the range of permitted values over which the search process may operate, and the number of objectives.

Results are promising: figure 1a depicts previously calibrated simulation behaviour (left), and NSGA-II's best attempt at recreating it (right) (Tripp, 2013). The absolute difference between target (hand-calibrated) values and those obtained by NSGA-II are shown in figure 1b. This experiment represents the most difficult problem setup that was attempted. NSGA-II operated on 6 objectives, attempting to match the peak numbers of CD4Th1, CD4Treg, CD8Treg, the times at which these peaks occurred for CD4Th1 and CD8Treg, and the number of CD4Th1 cells remaining at 40 days. The difficulty of this search problem must be emphasized, the 8 parameters being calibrated constitutes a substantial search space, and 6 objectives is a lot for a multi-objective optimization algorithm such as this. Nonetheless, the results are promising and further investigation, in particular incorporating dynamics from a second experiment, is warranted. Interestingly, the search process highlighted how disparate areas of search space could provide seemingly well-aligned behaviours (termed *local optima*) when calibrating against this one experiment. This highlights the importance of more powerful calibration techniques: the existence of multiple local optima in immunological simulations when calibrating against single experiments is problematic for reasons outlined above. Furthermore, if state-of-





(a) Effector T cell population sizes within the simulation over time. Left, the hand-calibrated simulation dynamics. Right, the result of NSGA-II calibrating ARTIMMUS parameters and recreating hand-calibrated dynamics

| response       | CD4Th1P | CD4TregP | CD8TregP | CD4Th1T | CD8TregT | CD4Th140 |
|----------------|---------|----------|----------|---------|----------|----------|
| abs difference | 47.0    | 76.0     | 55.0     | 5.0     | 0.0      | 0.0      |

(b) The absolute difference in values between hand calibrated response values, and those resulting from NSGA-II. Response names ending in 'P' denote peak population sizes, 'T' denotes the time at which these peaks occur.

| Parameter     | hand | NSGA-II (183) | range     | NSGA-LA (261) |
|---------------|------|---------------|-----------|---------------|
| CD4Th         | 40   | 30            | 0 - 100   | 43            |
| CD4Treg       | 30   | 38            | 0 - 90    | 55            |
| CD8Treg       | 30   | 47            | 0 - 90    | 50            |
| Neurons       | 500  | 527           | 440 - 560 | 460           |
| Microglia     | 75   | 106           | 15 - 135  | 117           |
| DCs in LN     | 10   | 13            | 0 - 70    | 43            |
| DCs in CNS    | 40   | 59            | 0 - 100   | 40            |
| DCs in Spleen | 100  | 72            | 40 - 160  | 109           |

(c) The parameters over which NSGA-II performs optimization. Hand calibrated and NSGA-II parameter values are given, as are the ranges of values over which NSGA-II operated. The best solution found is labelled 'NSGA-II'. DC, dendritic cells; LN, lymph node; CNS, central nervous system. A sub-optimal, but good, result from NSGA-II (labelled 'NSGA-LA') is also given. Fitnesses are shown in parentheses, and represent the sum of absolute differences between hand calibrated and NSGA-optimised response values. Lower fitnesses represent better solutions.

Figure 1: The result from NSGA-II that most closely calibrated ARTIMMUS parameter values against the hand-calibrated target values.

the-art automated methods with access to considerable computational power are challenged by this calibration problem, then those calibrating by eye and hand with trial and error ought to be cautious.

### Acknowledgements

MR and JT are supported by the EC CoCoRo Project, GA 270382. JT is also supported by the Royal Society.

### References

- K Deb, A Pratap, S. A. T. M. (2002). A fast and elitist multiobjective genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.
- Mark Read, Paul S. Andrews, J. T. V. K. (2012). Techniques for grounding agent-based simulations in the real domain: a case study in experimental autoimmune encephalomyelitis. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):67–86.
- Read, M. (2011). *Statistical and Modelling Techniques to Build Confidence in the Investigation of Immunology through Agent-Based Simulation*. PhD thesis, The University of York, UK.
- Tripp, M. (2013). Automated calibration of agent-based simulation for an autoimmune disease. Master's thesis, The University of York, UK.

# Enhancing the learning capacity of immunological algorithms: a comprehensive study of learning operators

Shangce Gao<sup>1</sup>, Tao Gong<sup>1</sup>, Weiya Zhong<sup>2</sup> and Fang Wang<sup>3</sup> and Beibei Chen<sup>1</sup>

<sup>1</sup>College of Information Science and Technology, Donghua University, China

<sup>2</sup>Department of Mathematics, Shanghai University, China

<sup>3</sup>Department of Information Systems and Computing, Brunel University, Uxbridge UB8 3PH, UK  
{gaosc,taogong}@dhu.edu.cn

## Abstract

Immunological algorithms are a kind of bio-inspired intelligence methods which draw inspiration from natural immune systems. The problem-solving performance of immunological algorithms mainly lies on the utilization of learning (i.e. mutation) operators. In this paper, nine different learning operators in a standard immune algorithmic framework are investigated. These learning operators consist of eight existing operators and a newly proposed search direction based operator. Experiments are conducted based on nine variants of immunological algorithms that use different learning operators. Simulation results on a large number of benchmark optimization problems give a deep insight into the characteristics of these operators, and further verify that the proposed new learning operator can greatly improve the performance of immunological algorithms.

## Introduction

Many difficulties such as dimensionality, differentiability and multimodality are associated with the optimization of large scale problems. To address such problems, bio-inspired intelligence algorithms (Da Silva Santos et al., 2010; Gao, 2012) have attracted more and more interest, among which the immunological algorithm (IA) is a particular class of optimization methods inspired by the basic features of adaptive immune response to antigenic stimulus. Most IAs mimic the metaphors of clonal selection principle (de Castro and Zuben, 2002), hypermutation (Freitas and Timmis, 2007), receptor editing (Gao et al., 2007) and lateral interaction effect (Whitbrook et al., 2007), providing a promising search mechanism by exploiting and exploring the solution space in parallel and effectively (Dasgupta et al., 2011). The main unique property of IAs is the utilization of the clonal proliferation, and the clonal selection which returns promising solutions acquired in the learning process. It is evident that IAs possess good features of maintaining population diversity, and capability of allocating multiple optimal solutions (Haktanirlar Ulutas and Kulturel-Konak, 2011). Although IAs have achieved good performance in solving various kinds of practical problems, such as digital signal processor (Mitra and Venayagamoorthy, 2010),

nonlinear classification (Ozsen et al., 2009), fault diagnosis (Hao and Sun, 2007), etc, their performance is limited in solving optimization problems (McEwan and Hart, 2009). Compared with other bio-inspired algorithms, such as the well-known evolutionary computation (Yao and Xu, 2006), IAs still greatly suffer from the issues of stagnation and slow convergence. The reason seems to be that the learning capacity (involving hypermutation and receptor editing) has not been fully exploited, i.e., no sophisticated learning operator can be found in the literature (Jansen and Zarges, 2011). Based on the above consideration, we review and analyze the existing learning operators commonly used in IAs, and propose a new search direction based learning operator ( $L_{sd}$ ) to encourage the antibodies to utilize the information of its surrounding antibodies, by means of moving the antibody toward the nearby antibodies with higher affinities and meanwhile away from the antibody with lower affinities. Therefore, the  $L_{sd}$  operator can not only evolve antibodies into promising search areas to accelerate convergence speed, but also prevent antibodies from entering undesired regions to jump out of local optimal solutions. The experiments of using all learning operators in IAs are conducted based on a large number of benchmark numerical optimization problems. The results show the characteristics of each learning operator, and further indicate that the proposed  $L_{sd}$  operator manipulates the best performance.

## Immunological algorithm

To investigate the effect of learning operators, a standard immunological algorithm framework (called IA) is utilized (de Castro and Zuben, 2002; Kelsey and Timmis, 2003). IA evolves a population of antibodies (B cells) towards a global optimum through a process of evaluation, cloning, learning (i.e. mutation) and selection. The evaluation procedure computes the affinity function values for all antibodies. Affinity is an important measure to represent the fitness of antibody to antigen. For a minimization optimization problem, higher affinity values of antibodies correspond to better solutions for the problem needed to be solved. The cloning proliferation is a mitotic procedure whereby the cells divide

themselves, creating a set of clones identical to the parent cell. Generally, the proliferation rate is directly proportional to the affinity level. The learning procedure (involving hypermutation or receptor editing) performs the exploration and exploitation within solution search space, guiding antibodies to the global optimum. It plays a key effect on the search performance of the algorithm. The selection procedure picks up the antibodies with higher affinities and meanwhile eliminates those with lower affinities to reduce the computational complexity.

From the perspective of optimization, a learning operator  $L$  mutate a candidate solution  $Ab = (x_1, x_2, \dots, x_D)$  to a trial one  $Ab' = L(Ab) = (x'_1, x'_2, \dots, x'_D)$ , where  $D$  is the dimension of the problem. In the literature, there are several learning operators commonly used in IA. They are summarized in the following.

- Gaussian mutation  $L_{gs}$  (Yao et al., 1999; de Castro and Timmis, 2002; Xu and Zhang, 2007; Khilwani et al., 2008; Song et al., 2006; Woldemariam and Yen, 2010):  $Ab' = Ab + \alpha N(0, 1)$ , where  $N(0, 1)$  is a random Gaussian number generated by the Gaussian function given as  $f_{Gaussian}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ , and  $\alpha$  controls the learning intensity imposed on the antibodies.
- Cauchy mutation  $L_{cy}$  (Yao et al., 1999; Xu and Zhang, 2007; Khilwani et al., 2008):  $Ab' = Ab + \alpha \delta_k$ , where  $\delta_k$  is a Cauchy random variable with the scale parameter  $t = 1$  and satisfies the density function  $f_{Cauchy}(x) = \frac{t}{\pi(t^2 + x^2)}$ .
- Static Hypermutation  $L_{h1}$  (Cutello et al., 2004; Gong et al., 2008): the number of mutations is independent from the affinity of the antibody. That is to say,  $Ab'$  will undergo a constant number  $c$  of mutation times. Each mutation act on  $Ab$  is implemented through replacing a certain number of  $Ab$  at a random dimension with a random integer between 0 and 9 (Gong et al., 2008).
- Proportional Hypermutation  $L_{h2}$  (Cutello et al., 2004): the number of mutations is proportional to the normalized affinity value, that is,  $\hat{f}(Ab) \times c \times D$ , where  $\hat{f}(Ab)$  is the normalized affinity distributed in the interval of  $[0, 1]$ .  $c$  is a constant number, representing the maximum mutation intensity.
- Inversely proportional hypermutation  $L_{h3}$  (Cutello et al., 2004, 2005, 2006): the number of mutations is inversely proportional to the normalized affinity value, i.e., the higher affinity of an antibody, the less times of mutations will be carried out on it. It is reasonable to make such an inverse choice, since better antibodies usually contain more useful information for evolution. Too many mutations might have higher probability to destroy these information, thus depressing the learning performance.
- Hypermacromutation  $L_m$  (Cutello et al., 2004): the number of mutations is independent from the affinity and the parameter  $c$ . Instead, the operator mutates at most  $j - i + 1$  values in the interval of  $[i, j]$ , where two randomly generated integers  $i$  and  $j$  satisfy the condition of  $i < j \leq D$ .
- Lateral interaction mutation  $L_{li}$  (Cutello et al., 2006; Pavone et al., 2011): in addition to hypermutation and receptor editing, the lateral interaction during different antibodies also takes place according to the idiotypic network theory (Gao et al., 2008). In other words, each paratope on an antibody can not only recognize a foreign antigen, but also can be recognized by external idiotopes. Motivated by this mechanism, similar to the crossover operator in evolutionary computation, an antibody is attracted by other antibodies, i.e.,  $Ab'_i = (1 - \beta) \times Ab_i + \beta Ab_j$ , where  $Ab_j \neq Ab_i$  is a randomly selected antibody in the population.
- Baldwinian learning  $L_{bl}$  (Gong et al., 2010): learning mechanism can provide an easy evolutionary path towards co-adapted alleles in environments, by means of employing differential information during other antibodies. It is realized as  $Ab'_i = Ab_i + s \times (Ab_j - Ab_k)$  in a probability of  $p$ , where  $i \neq j \neq k$ ,  $Ab_j$  and  $Ab_k$  are randomly selected from the population, and  $s$  represents the Baldwinian learning strength.

Intuitively, all the above eight learning operators are able to evolve antibodies to matured ones in semi-blind manners, although some of the matured ones might possess lower affinities. However, as the parallel feature of the immune algorithm, there did exist a probability of making progress to improve the affinity of antibodies. After the clonal selection progress, the most improved antibodies are reserved and enter into the next generation of evolution.

### Search direction based learning operator

Even though the above learning operators used in IA can explore/exploit the solution space in an effective manner, as we observed, they are not fully developed from the aspect of utilizing the information in environment. In Fig. 1, we summarized the characteristics of the learning operators. The solid rectangle  $S$  show the solution space of the optimization problem. The dashed circles denote contour lines of affinity, and the inner circles indicate that they represent higher affinities than the outer ones.

From Fig. 1, we can notice that the learning mechanisms used in (1)-(6) on the antibody  $Ab$  only utilize random perturbation on the antibody itself, while those in (7)-(8) make use of information in the environment. As reported in (Cutello et al., 2006; Gong et al., 2010), learning from the environment provides an encouraging alternative method, probably a more easy way to achieve better search performance. In details, the mechanism in (7) uses the information of a randomly selected antibodies in the population to

guide the current search. A successful guide is strongly depending on the quality of the selected guiding antibody  $Ab_j$ , implying that there must be amount of redundant search if the guiding antibody is far away from the global optimal solution. Instead, the mechanism in (8) utilizes the differential information between two other antibodies  $Ab_j$  and  $Ab_k$  in the population. The learning acting on this differential information might have ability to use the mutual beneficial components, thus exhibiting more promising properties for searching.

Based on the above analysis, we can find that the mechanisms used in (7) and (8) don't use any measurable knowledge from the population, i.e., the guiding antibodies ( $Ab_j$  and  $Ab_k$ ) are randomly selected without any relevance of the current antibody  $Ab$ , therefore hindering the effectiveness of the learning performance. In view of the limitations of the above learning operators, we propose a new search direction based learning operator  $L_{sd}$ , not only to provide another alternative mutation method, but also aiming to achieve a better search capacity. The  $L_{sd}$  is formulated in Eqs. (1)-(3):

In these equations,  $\Delta_{repulsion}$  and  $\Delta_{attraction}$  are repulsion and attraction effect on  $Ab_i$  respectively. In a single learning procedure of  $L_{sd}$ , if a randomly selected guiding antibody  $Ab_j$  whose affinity is lower than the base antibody  $Ab_i$ , then the repulsion effect will be implemented with a probability of  $p_r$ , for the purpose of preventing the base antibody from entering undesired region of the search space. On the other hand, the attraction effect takes place with a probability of  $p_a$  when the affinity of the selected guiding antibody  $Ab_k$  is higher, thus enhancing the capacity of exploiting the promising regions of the search space. In addition, to improve the randomness of the learning mechanism, the attraction and repulsion scaling factors  $\alpha_1$  and  $\alpha_2$  are set as random numbers generated in the interval  $[0, 1]$ . From Eq. (3), it is clear that the search direction of the base antibody is guided to move towards the regions with higher affinity, and meanwhile away from the regions with lower affinity, thus enabling the algorithm to possess better exploitation capacity and the ability of jumping out of local optimum.

## Experimental results and discussions

The computational progresses of nine learning operators used in IA described above have been implemented in C++ program under Visual Studio 2010. In order to evaluate the performance of the proposed  $L_{sd}$  learning operator, it is validated using some well-known benchmark numerical optimization problems obtained from the literatures (Yao et al., 1999; Cutello et al., 2006; Gong et al., 2010). Table 1 lists the details of the benchmark functions.  $f_1 - f_5$  are unimodal functions which are relatively easy to be optimized, but the difficulty increases as the dimension size increases.  $f_6$  is the step function, while  $f_7$  is a noisy quartic function.  $f_8 - f_{13}$  are multimodal functions with plenty of local minima which represent the most difficult class of problems for many op-

timization algorithms;  $f_{14} - f_{23}$  is a multimodal function with only a few local optima. The different type of benchmark functions test the searching ability of learning operators from different aspects, that is, unimodal functions tend to reflect the convergence speed of the operator in a direct manner, while multimodal functions are likely to estimate the operator's capacity of escaping from local optima.

Owing to the random nature of the IA and the learning operators, to evaluate the performance of each learning operator, their performance cannot be judged by the result of a single run. Many trials with independent population initialization should be made to obtain a useful conclusion of the performance of the approach. Therefore, in this study the results are obtained in 30 trials. In the experiment, the user-defined parameters are set as follows: the population size is set to be 30, the clone size is 5. It is worth mentioning that we use equal cloning strategy in this study to reduce the influences of cloning operator. By doing so, each antibody in the population has the same probability to undergo the learning mechanism, thus we can make a direct comparison of the performance during all learning operators. In addition, the termination condition of the algorithm is set to be that when the maximum number of function evaluations reaches 150000. Fig. 2 depicts the sketch of the Sphere function  $f_1$  when its dimension  $D$  is set to be 2, and the corresponding convergence graphs of each learning operator. It is obvious that, for such unimodal function, all learning operators can evolve the antibodies effectively. In particular, the learning operator  $L_{sd}$  possesses the fast learning speed and the most precise solution.

To further demonstrate the effectiveness and robustness of the proposed  $L_{sd}$ , all learning operators are carried out on all tested 23 benchmark functions. One of the effective strategies to perform a comparative study between the variants of IAs is to use the oracle-based view of computation (Wolpert and Macready, 1997). Based on this method, the best solution should be found within a certain number of function evaluations. Herein, the best values can be used for comparison because of the equal number of function evaluation for all operators in all cases. To make an intuitive comparison during the variants, the results of best run are normalized between 0 and 1, therefore the worst and best values of each best solution are changed to 0 and 1, respectively. The normalized results for all benchmark functions are presented in Table 2. To achieve a general conclusion based on the oracle-based view, three kinds of the sum of scores and rank of each learning operator are presented in this table. The symbol  $\sum_u$  denotes the sum of scores on unimodal function  $f_1 - f_5$ , while  $\sum_m$  is the sum on multimodal functions  $f_8 - f_{13}$ .  $\sum$  represents the total sum on all tested benchmark functions. At first glance, it is clear that the learning operator  $L_{sd}$  works very well because it has the best performance with the score of 22.940 to 23, and has a rank of 1 among 9 operators. Furthermore, as the  $\sum_u$  of



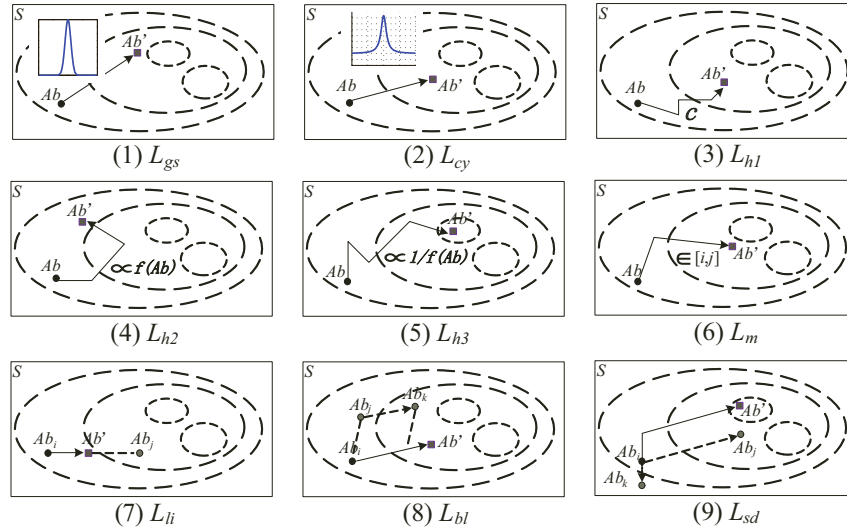


Figure 1: The learning characteristics of all nine operators.

$L_{sd}$  is 4.941 which is also the best one among all operators, thus confirming that  $L_{sd}$  has a fastest convergence speed for unimodal functions. On the other hand, the biggest value of  $\sum_m$  of  $L_{sd}$  also verifies the capability of escaping from local optimum for multimodal functions.

Based on Table 2, there are some remarks should be emphasized concerning all learning operators. The first six learning operators mutate the base antibody only utilizing random perturbation, while the last three ones make use of information in the population either semi-blind or search direction based. The population information utilization based learning operators have ranks of 1, 2, 3, significant better than the others, suggesting that the interaction of information in the population is likely to improve the search performance. Thus, we can conclude that the former operators mainly act as the exploitation in the search space, while the latter ones mainly employed as exploration. In the further, it is a promising research direction to combine one of former six operators with the one of latter ones, and it can be expected to achieve a better performance.

## Conclusions

In this paper, we made a comprehensive study on the learning operators used in the immunological algorithms. Nine different learning operators, maturing the antibody either by utilizing random perturbation or by utilizing the guiding information from other antibodies, are implemented and analyzed. In view of the limitations of the existing operators, the newly proposed search direction based learning mechanism can not only attract the antibody to promising regions in search space, but also preventing from entering undesired regions by means of the information contained in the antibodies with worse affinities. Experimental results on a large number of benchmark numerical optimization problems verified the effectiveness and robustness of  $L_{sd}$ , suggesting that the useful information during the whole population should be sufficiently utilized to improve the search performance of the algorithm.

## Acknowledgements

This work is partially supported by the National Natural Science Foundation of China under Grants 61203325, 61271114 and 61003205, Genguang Project of Shanghai E-

$$\Delta_{repulsion} = \begin{cases} Ab_j - Ab_i, & \text{if } rand() < p_r \text{ for } \forall f(Ab_j) < f(Ab_i) \\ Ab_i, & \text{otherwise} \end{cases} \quad (1)$$

$$\Delta_{attraction} = \begin{cases} Ab_k - Ab_i, & \text{if } rand() < p_a \text{ for } \forall f(Ab_k) \geq f(Ab_i) \\ Ab_i, & \text{otherwise} \end{cases} \quad (2)$$

$$Ab' = Ab + \alpha_1 \times \Delta_{attraction} - \alpha_2 \times \Delta_{repulsion} \quad (3)$$

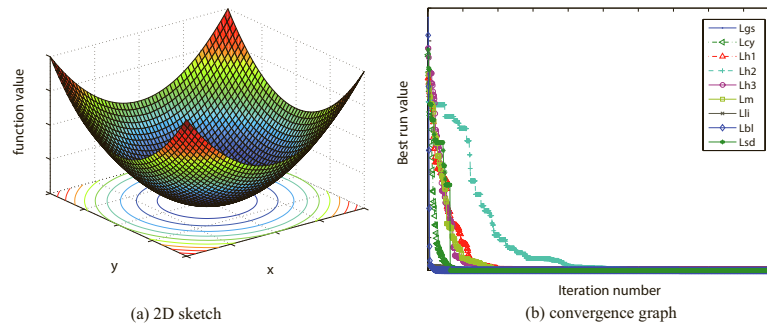


Figure 2: (a) the 2D sketch of the Sphere function  $f_1$ , (b) the convergence graphs of each learning operators on  $f_1$ .

educational Development Foundation (No. 12CG35), Ph.D. Program Foundation of Ministry of Education of China (No. 20120075120004), and the Fundamental Research Funds for the Central Universities.

### References

- Cutello, V., Narzisi, G., Nicosia, G., and Pavone, M. (2005). An immunological algorithm for global numerical optimization. In *Proceedings of the 7th international conference on Artificial Evolution*, pages 284–295. Springer-Verlag.
- Cutello, V., Nicosia, G., and Pavone, M. (2004). Exploring the capability of immune algorithms: A characterization of hypermutation operators. In *Third International Conference on Artificial Immune Systems, ICARIS 2004*, pages 263–276.
- Cutello, V., Nicosia, G., and Pavone, M. (2006). Real coded clonal selection algorithm for unconstrained global optimization using a hybrid inversely proportional hypermutation operator. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 950–954. ACM.
- Da Silva Santos, C. H., Goncalves, M. S., and Hernandez-Figueroa, H. E. (2010). Designing novel photonic devices by bio-inspired computing. *IEEE Photonics Technology Letters*, 22(15):1177–1179.
- Dasgupta, D., Yu, S., and Nino, F. (2011). Recent advances in artificial immune systems: Models and applications. *Applied Soft Computing*, 11:1574–1587.
- de Castro, L. and Zuben, F. J. V. (2002). Learning and optimization using clonal selection principle. *IEEE Trans. on Evolutionary Computation*, 6(3):239–251.
- de Castro, L. N. and Timmis, J. (2002). An artificial immune network for multimodal function optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02*, volume 1, pages 699–704. IEEE.
- Freitas, A. A. and Timmis, J. (2007). Revisiting the foundations of artificial immune systems for data mining. *IEEE Trans. on Evolutionary Computation*, 11(4):521–540.
- Gao, S. C., editor (2012). *Bio-Inspired Computational Intelligence and Their Applications*. INTECH press, New York, USA.
- Gao, S. C., Dai, H., Zhang, J., and Tang, Z. (2008). An expanded lateral interactive clonal selection algorithm and its application. *IEICE Trans. Fundamentals*, E91-A(8):2223–2231.
- Gao, S. C., Tang, Z., Dai, H., and Zhang, J. (2007). An improved clonal algorithm and its application to traveling salesman problems. *IEICE Trans. Fundamentals*, E90-A(12):2930–2938.
- Gong, M., Jiao, L., and Zhang, L. (2010). Baldwinian learning in clonal selection algorithm for optimization. *Information Sciences*, 180(8):1218–1236.
- Gong, M., Jiao, L., and Zhang, X. (2008). A population-based artificial immune system for numerical optimization. *Neurocomputing*, 72(1-3):149–161.
- Haktanirlar Ulutas, B. and Kulturel-Konak, S. (2011). A review of clonal selection algorithm and its applications. *Artificial Intelligence Review*, 36(2):117–138.
- Hao, X. and Sun, C. (2007). Artificial immune network classification algorithm for fault diagnosis of power transformer. *IEEE Trans. on Power Delivery*, 22(2):930–935.
- Jansen, T. and Zarges, C. (2011). Analyzing different variants of immune inspired somatic contiguous hypermutations. *Theoretical Computer Science*, 412(6):517–533.
- Kelsey, J. and Timmis, J. (2003). Immune inspired somatic contiguous hypermutation for function optimisation. In *Genetic and Evolutionary Computation, GECCO 2003*, pages 207–212. Springer.
- Khilwani, N., Prakash, A., Shankar, R., and Tiwari, M. K. (2008). Fast clonal algorithm. *Engineering Applications of Artificial Intelligence*, 21:106–128.
- McEwan, C. and Hart, E. (2009). On aircs and clonal selection for machine learning. In *Artificial Immune Systems, Artificial Immune Systems*, pages 67–79. Springer.
- Mitra, P. and Venayagamoorthy, G. K. (2010). An adaptive control strategy for dstatcom applications in an electric ship power system. *IEEE Trans. on Power Electronics*, 25(1):95–104.
- Ozsen, S., Gunes, S., Kara, S., and Latifoglu, F. (2009). Use of kernel functions in artificial immune systems for the nonlinear classification problems. *IEEE Trans. on Information Technology in Biomedicine*, 13(4):621–628.

- Pavone, M., Narzisi, G., and Nicosia, G. (2011). Clonal selection: an immunological algorithm for global optimization over continuous spaces. *Journal of Global Optimization*, pages 1–40.
- Song, Z. Y., Gao, X. Z., Hiang, X. L., and Lin, H. S. (2006). A modified immune optimization algorithm. In *International Conference on Machine Learning and Cybernetics*, pages 2184–2189. IEEE.
- Whitbrook, A. M., Aickelin, U., and Garibaldi, J. (2007). Idiotypic immune networks in mobile-robot control. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(6):1581–1598.
- Woldemariam, K. M. and Yen, G. G. (2010). Vaccine-enhanced artificial immune system for multimodal function optimization. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(1):218–228.
- Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Trans. on Evolutionary Computation*, 1:67–82.
- Xu, X. and Zhang, J. (2007). An improved immune evolutionary algorithm for multimodal function optimization. In *Third International Conference on Natural Computation, ICNC 2007*, volume 3, pages 641–646. IEEE.
- Yao, X., Liu, Y., and Lin, G. (1999). Evolutionary programming made faster. *IEEE Trans. on Evolutionary Computation*, 3(2):82–102.
- Yao, X. and Xu, Y. (2006). Recent advances in evolutionary computation. *Journal of Computer Science and Technology*, 21(1):1–18.

Table 1: Benchmark problems used in the experiments.

| Function Definition |                                                                                                                                                                                                                                                                                                                                 | Dim. | Domain $S$                | $f_{min}$  |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------------------|------------|
| $f_1(X)$            | $= \sum_{i=1}^n x_i^2$                                                                                                                                                                                                                                                                                                          | 30   | $[-100, 100]^n$           | 0          |
| $f_2(X)$            | $= \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $                                                                                                                                                                                                                                                                                    | 30   | $[-10, 10]^n$             | 0          |
| $f_3(X)$            | $= \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$                                                                                                                                                                                                                                                                                           | 30   | $[-100, 100]^n$           | 0          |
| $f_4(X)$            | $= \max\{ x_i , 1 \leq i \leq n\}$                                                                                                                                                                                                                                                                                              | 30   | $[-100, 100]^n$           | 0          |
| $f_5(X)$            | $= \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$                                                                                                                                                                                                                                                                     | 30   | $[-30, 30]^n$             | 0          |
| $f_6(X)$            | $= \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$                                                                                                                                                                                                                                                                                  | 30   | $[-100, 100]^n$           | 0          |
| $f_7(X)$            | $= \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$                                                                                                                                                                                                                                                                                   | 30   | $[-1.28, 1.28]^n$         | 0          |
| $f_8(X)$            | $= \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$                                                                                                                                                                                                                                                                                        | 30   | $[-500, 500]^n$           | -12569.5   |
| $f_9(X)$            | $= \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$                                                                                                                                                                                                                                                                               | 30   | $[-5.12, 5.12]^n$         | 0          |
| $f_{10}(X)$         | $= -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2})$                                                                                                                                                                                                                                                                        |      |                           |            |
|                     | $-\exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$                                                                                                                                                                                                                                                                       | 30   | $[-32, 32]^n$             | 0          |
| $f_{11}(X)$         | $= \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$                                                                                                                                                                                                                                            | 30   | $[-600, 600]^n$           | 0          |
| $f_{12}(X)$         | $= \frac{\pi}{n} \{10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})]$<br>$+ (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4),$<br>$y_i = 1 + \frac{1}{4}(x_i + 1)$<br>$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$ | 30   | $[-50, 50]^n$             | 0          |
| $f_{13}(X)$         | $= 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1) [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$                                                                                                                                                                    | 30   | $[-50, 50]^n$             | 0          |
| $f_{14}(X)$         | $= [\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}]^{-1}$                                                                                                                                                                                                                                          | 2    | $[-65.536, 65.536]^n$     | 1          |
| $f_{15}(X)$         | $= \sum_{i=1}^{11} [a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$                                                                                                                                                                                                                                                | 4    | $[-5, 5]^n$               | 0.0003075  |
| $f_{16}(X)$         | $= 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$                                                                                                                                                                                                                                                             | 2    | $[-5, 5]^n$               | -1.0316285 |
| $f_{17}(X)$         | $= (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi} \cos x_1 + 10$                                                                                                                                                                                                                              | 2    | $[-5, 10] \times [0, 15]$ | 0.398      |
| $f_{18}(X)$         | $= [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times$<br>$[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$                                                                                                                                                               | 2    | $[-2, 2]^n$               | 3          |
| $f_{19}(X)$         | $= -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2]$                                                                                                                                                                                                                                                                | 3    | $[0, 1]^n$                | -3.86      |
| $f_{20}(X)$         | $= -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2]$                                                                                                                                                                                                                                                                | 6    | $[0, 1]^n$                | -3.32      |
| $f_{21}(X)$         | $= -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$                                                                                                                                                                                                                                                                             | 4    | $[0, 10]^n$               | -10.1422   |
| $f_{22}(X)$         | $= -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$                                                                                                                                                                                                                                                                             | 4    | $[0, 10]^n$               | -10.3909   |
| $f_{23}(X)$         | $= -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$                                                                                                                                                                                                                                                                          | 4    | $[0, 10]^n$               | -10.53     |



Table 2: Normalized statistical results of learning operators,  $L_{gs}$ ,  $L_{cy}$ ,  $L_{h1}$ ,  $L_{h2}$ ,  $L_{h3}$ ,  $L_m$ ,  $L_{li}$ ,  $L_{bl}$ ,  $L_{sd}$  for the benchmark problems.

| Function | $L_{gs}$ | $L_{cy}$ | $L_{h1}$ | $L_{h2}$ | $L_{h3}$ | $L_m$  | $L_{li}$ | $L_{bl}$ | $L_{sd}$ |
|----------|----------|----------|----------|----------|----------|--------|----------|----------|----------|
| $f_1$    | 0.856    | 0.901    | 0.285    | 0.000    | 0.800    | 0.915  | 1.000    | 1.000    | 1.000    |
| $f_2$    | 0.000    | 0.785    | 0.567    | 0.234    | 0.965    | 0.865  | 0.999    | 1.000    | 1.000    |
| $f_3$    | 0.000    | 0.235    | 0.245    | 0.345    | 0.657    | 0.768  | 0.987    | 1.000    | 1.000    |
| $f_4$    | 0.000    | 0.324    | 0.156    | 0.000    | 0.483    | 0.481  | 0.925    | 1.000    | 0.956    |
| $f_5$    | 0.000    | 0.124    | 0.000    | 0.235    | 0.210    | 0.454  | 0.768    | 0.923    | 0.985    |
| $f_6$    | 0.125    | 0.248    | 0.000    | 0.000    | 0.358    | 0.405  | 0.567    | 0.999    | 1.000    |
| $f_7$    | 0.450    | 0.500    | 0.000    | 0.000    | 0.146    | 0.056  | 0.679    | 0.956    | 1.000    |
| $f_8$    | 0.235    | 0.167    | 0.250    | 0.000    | 0.580    | 0.375  | 0.788    | 0.876    | 1.000    |
| $f_9$    | 0.000    | 0.056    | 0.120    | 0.000    | 0.450    | 0.734  | 0.567    | 0.752    | 1.000    |
| $f_{10}$ | 0.125    | 0.450    | 0.045    | 0.000    | 0.236    | 0.678  | 0.458    | 1.000    | 0.999    |
| $f_{11}$ | 0.250    | 0.467    | 0.000    | 0.011    | 0.235    | 0.385  | 0.572    | 0.750    | 1.000    |
| $f_{12}$ | 0.000    | 0.450    | 0.245    | 0.000    | 0.560    | 0.476  | 0.877    | 0.999    | 1.000    |
| $f_{13}$ | 0.258    | 0.782    | 0.000    | 0.000    | 0.450    | 0.359  | 0.578    | 0.974    | 1.000    |
| $f_{14}$ | 0.856    | 0.978    | 0.450    | 0.000    | 0.874    | 0.385  | 0.683    | 0.999    | 1.000    |
| $f_{15}$ | 0.784    | 0.654    | 0.000    | 0.000    | 0.487    | 0.530  | 0.976    | 0.965    | 1.000    |
| $f_{16}$ | 0.460    | 0.750    | 0.000    | 0.674    | 0.576    | 0.045  | 0.956    | 0.999    | 1.000    |
| $f_{17}$ | 0.470    | 0.865    | 0.000    | 0.000    | 0.430    | 0.012  | 0.995    | 0.999    | 1.000    |
| $f_{18}$ | 0.956    | 1.000    | 0.385    | 0.000    | 0.450    | 0.755  | 0.999    | 0.999    | 1.000    |
| $f_{19}$ | 0.845    | 0.999    | 0.568    | 0.000    | 0.450    | 0.785  | 1.000    | 1.000    | 1.000    |
| $f_{20}$ | 0.864    | 0.968    | 0.452    | 0.000    | 0.969    | 0.704  | 0.933    | 0.999    | 1.000    |
| $f_{21}$ | 0.765    | 0.742    | 0.000    | 0.011    | 0.345    | 0.358  | 1.000    | 1.000    | 1.000    |
| $f_{22}$ | 0.875    | 0.785    | 0.075    | 0.000    | 0.550    | 0.340  | 1.000    | 1.000    | 1.000    |
| $f_{23}$ | 0.920    | 0.965    | 0.105    | 0.000    | 0.568    | 0.285  | 1.000    | 1.000    | 1.000    |
| $\sum^u$ | 0.856    | 2.369    | 1.253    | 0.814    | 3.115    | 3.483  | 4.679    | 4.923    | 4.941    |
| $\sum^m$ | 0.868    | 2.372    | 0.660    | 0.011    | 2.511    | 3.007  | 3.840    | 5.351    | 5.999    |
| $\sum$   | 10.094   | 14.195   | 3.948    | 1.510    | 11.829   | 11.150 | 19.307   | 22.189   | 22.940   |
| Rank     | 7        | 4        | 8        | 9        | 5        | 6      | 3        | 2        | 1        |

# Real-Valued Negative Databases

Dongdong Zhao<sup>1,2</sup> and Wenjian Luo<sup>1,2,\*</sup>

<sup>1</sup> School of Computer Science and Technology,

University of Science and Technology of China, Hefei 230027, Anhui, China

<sup>2</sup> Anhui Province Key Laboratory of Software Engineering in Computing and Communication,

University of Science and Technology of China, Hefei 230027, Anhui, China

zdd@mail.ustc.edu.cn, wjluo@ustc.edu.cn

## Abstract

The negative database (*NDB*) is the negative representation of original data. Existing work has demonstrated that *NDB* can be used to preserve privacy and hide information. However, most work about *NDB* is based on binary representation. In some applications which are naturally described in real-valued space, the binary negative database is hard to be applied appropriately. Therefore, the real-valued negative database is proposed in this paper, and reversing the real-valued negative database is proved to be an *NP*-hard problem. Moreover, an effective algorithm for generating real-valued negative databases is given. Finally, an example of applying the real-valued negative database to the privacy-preserving data publication is described, and it shows that the real-valued negative database is valuable in practice.

## Introduction

Nowadays, databases have become basic tools for storing data. As the privacy of data is widely concerned, the techniques which can preserve privacy while keeping the database services available are urgently needed. Traditional databases store the data with the form what it actually is. This way is called the positive representation of data, and the databases are called positive databases. The privacy of traditional databases is easy to be revealed when the databases are leaked. Although some cryptography methods can be applied to the positive databases, it is time-consuming to encrypt every entry in the databases and the encrypted databases cannot support basic database operations efficiently. Another way is to control the access of the positive database, but this way cannot eliminate all the security risks as there may be some internal attacks.

The negative database, which is inspired by Natural Immune System, was proposed by Esponda and his colleagues (Esponda et al., 2004a; Esponda et al., 2004b; Esponda et al., 2005; Esponda et al., 2007a; Esponda et al., 2009). In contrast to traditional databases, the negative database only stores the information in the complementary set of the original data. This way is called the negative representation of data. It has been proved that reversing the negative database with the binary representation (i.e. recovering the corresponding binary

positive database) is *NP*-hard (Esponda et al., 2004b; Esponda et al., 2009). Therefore, the binary negative database could be employed to protect data privacy. Some algorithms for generating binary negative databases from binary positive databases have been proposed, such as the prefix algorithm (Esponda et al., 2004b; Esponda et al., 2009), the *RNDB* algorithm (Esponda et al., 2004b; Esponda et al., 2009), the *q*-hidden algorithm (Jia et al., 2005; Esponda et al., 2007a) and the hybrid-*NDB* algorithm (Liu et al., 2011). Furthermore, some basic operations upon the negative database have been proposed, such as the negative Cartesian product, negative join and negative intersection (Esponda et al., 2004a; Esponda et al., 2005; Esponda et al., 2007b).

So far, most work about the negative database is based on the binary representation. However, in some applications which are naturally described in real-valued space, the negative database with the binary representation is not appropriate. Therefore, the real-valued negative database is proposed in this paper.

The negative database has already been introduced to some applications such as privacy preserving (Esponda et al., 2004b; Esponda et al., 2007a; Esponda et al., 2009), sensitive data collection (Esponda, 2006; Horey et al., 2007) and authentication (Dasgupta and Azeem, 2007; Dasgupta and Azeem, 2008). In this paper, an example of applying the real-valued negative database to the privacy-preserving data publication is given. This example demonstrates that the real-valued negative database is appropriate for the privacy-preserving data publication.

## Existing Work about Negative Databases

The negative database (*NDB*) was proposed by Esponda and his colleagues (Esponda et al., 2004a; Esponda et al., 2004b). Presently, most negative databases are based on the binary representation. The details of the binary negative database are described as follows (Esponda et al., 2004b; Esponda et al., 2009).

Assume the original data is a database which consists of  $n$  entries, i.e.  $DB = \{x_1, x_2, \dots, x_n\}$ , and each entry in  $DB$  is a binary string with length  $m$ . The universal set is  $U = \{0, 1\}^m$ .

\* Corresponding author. Tel: 86-551-63602824

The complementary set of  $DB$  is denoted as  $U-DB$ , and the negative database  $NDB$  only stores the elements that belong to  $U-DB$ . As there are usually too many binary strings belong to  $NDB$ , a “don’t care” symbol ‘\*’ is introduced to compress  $NDB$  to a reasonable size. Each entry in  $NDB$  is a string defined upon the alphabet  $\{0, 1, *\}$  with length  $m$ . The positions with value 0 or 1 are called specified positions, and those with symbol ‘\*’ are called unspecified positions. The symbol ‘\*’ represents 0 or 1 at a given position. If all the entries in  $U-DB$  are covered by  $NDB$ ,  $NDB$  is said to be complete. Any binary string  $s$  is said to be matched with (or covered by) an entry  $y$  in  $NDB$  if and only if the value at each position of  $s$  is identical to that of  $y$  or the corresponding value of  $y$  is ‘\*’. With the unspecified value ‘\*’, multiple different negative databases can be generated from the same positive database.

It has been proved that reversing the binary negative database (i.e. recovering the corresponding binary positive database) is an  $NP$ -hard problem (Esponda et al., 2004b; Esponda et al., 2009). If reversing a negative database is computationally infeasible, the negative database is said to be hard-to-reverse, otherwise it is said to be easy-to-reverse.

Some algorithms for generating the binary negative database from a binary positive database have been proposed. The prefix algorithm (Esponda et al., 2004b; Esponda et al., 2009) is the first algorithm for generating binary negative databases, and it is compact and efficient. The binary negative database generated by the prefix algorithm is complete but easy-to-reverse. In order to overcome this shortcoming, the  $RNDB$  algorithm (Esponda et al., 2004b; Esponda et al., 2009) was proposed. The  $RNDB$  algorithm embeds some random factors for generating binary negative databases which are possibly hard-to-reverse. However, the hard-to-reverse property of the binary negative databases generated by the  $RNDB$  algorithm could not be guaranteed, and the size of those binary negative databases could be too large. The  $q$ -hidden algorithm (Jia et al., 2005; Esponda et al., 2007a) was proposed for the binary positive databases that contain only one entry, and it is very efficient. The generated binary negative databases are not complete, but hard-to-reverse on average. The hybrid- $NDB$  algorithm (Liu et al., 2011) combines the prefix algorithm with the  $q$ -hidden algorithm to generate binary negative databases that are both complete and hard-to-reverse on average. It is noted that the “hard-to-reverse” property mentioned here means that the SAT solvers with local search strategy (e.g. WalkSAT (Selman et al., 1995)) could not reverse the negative databases on average.

In real-world applications, real-valued databases are often used. However, it is not convenient to employ the binary negative database to represent a real-valued database. Therefore, the real-valued negative database is studied in this paper. It is noted that earlier work about the negative database is the negative selection algorithm (Forrest et al., 1994; Ji and Dasgupta, 2007). The binary negative database is closely related to the negative selection algorithm with the binary representation (Forrest et al., 1994; Ji and Dasgupta, 2007), while their objectives and generation algorithms are obviously different. Hence, the real-valued negative database is also related to (but different from) the negative selection algorithm with the real-valued representation (González et al., 2003; Ji

and Dasgupta, 2004; Ji and Dasgupta, 2006; Ji and Dasgupta, 2007).

## The Real-Valued Negative Database

Assume real-valued positive database ( $DB$ ) contains  $n$  entries, i.e.  $DB = \{x_1, x_2, \dots, x_n\}$ . There are  $m$  attributes  $\{R_1, R_2, \dots, R_m\}$  in  $DB$ , and the domain of each attribute  $R_k$  ( $k = 1 \dots m$ ) is  $I_k = [l_k, u_k]$ .  $l_k$  is the lower bound and  $u_k$  is the upper bound. The bounds  $l_k$  and  $u_k$  are both real values, i.e.  $l_k \in R$ ,  $u_k \in R$ . Each entry  $x_i$  ( $i = 1 \dots n$ ) is a vector of  $m$  real values, and each value  $x_i[k]$  ( $k = 1 \dots m$ ) belongs to the domain of  $k^{th}$  attribute, i.e.  $x_i[k] \in I_k$ .

The real-valued negative database only stores the information that belongs to the complementary set of the real-valued positive database. Since the instances covered by the real-valued negative database are usually too many to be presented exactly, intervals are introduced to compress them.

Suppose  $a$  is an entry with  $m$  real values, and  $v$  is an entry with  $m$  intervals. Entry  $a$  is matched with (or covered by) entry  $v$  if and only if following condition is satisfied.

$$a[k] \in v[k], \quad k = 1, 2, \dots, m \quad (1)$$

Based on above matching rule, the real-valued negative database for  $DB$  can be defined as follows.

**Definition 1.** (*Real-Valued Negative Database*) Giving the real-valued positive database  $DB$  and the universal set  $U = I_1 \times I_2 \times \dots \times I_m$ , the real-valued negative database ( $RvNDB$ ) for  $DB$  is a compressed representation of  $U-DB$ . Each entry in  $RvNDB$  consists of  $m$  intervals, and does not cover any entries in  $DB$ .

If  $RvNDB$  covers the whole complementary set of  $DB$ ,  $RvNDB$  is said to be complete. Otherwise,  $RvNDB$  is said to be incomplete. A simple database query can be processed directly upon the real-valued negative database. For any  $s$  (a vector with  $m$  real values), if it is covered by  $RvNDB$ , it does not belong to  $DB$ ; if  $s$  is not covered by  $RvNDB$  and  $RvNDB$  is complete, it belongs to  $DB$ .

As any two entries in the real-valued negative database may intersect with each other, one real-valued positive database can be mapped to multiple real-valued negative databases. An example is given in table 1.

**Table 1.** An example of  $RvNDB$ s

| $DB$     | $NDB_1$               | $NDB_2$                 |
|----------|-----------------------|-------------------------|
| 0.2, 0.8 | [0, 0.2), [0, 1.0]    | [0, 0.2), [0.8, 1.0]    |
|          | [0.21, 1.0], [0, 1.0] | [0.21, 1.0], [0.8, 1.0] |
|          | [0, 1.0], [0, 0.8]    | [0, 1.0], [0, 0.8]      |
|          | [0, 1.0], [0.81, 1.0] | [0, 1.0], [0.81, 1.0]   |

Notes: There are two attributes in  $DB$ . The domains of the two attributes are both  $[0, 1.0]$ .

## The $NP$ -Hard Property of $RvNDB$ s

In this section, reversing the real-valued negative database ( $RvNDB$ ) is proved to be an  $NP$ -hard problem. The proofs are

similar to the work in (Esponda et al., 2004b; Esponda et al., 2009). Based on the hardness of reversing the real-valued negative database, the real-valued negative database can be used to preserve privacy.

**Problem 1.** Is the positive database of  $RvNDB$  non-empty? That is, is there any entry that is not covered by  $RvNDB$ ?

**Problem 2.** Can  $RvNDB$  be reversed to obtain the entries in the corresponding positive database? That is, can any entry that is not covered by  $RvNDB$  be found?

**Lemma 1.** Problem 1 is *NP*.

*Proof.* Giving any entry  $w$  which consists of  $m$  real values, if an algorithm can check whether it is the solution in polynomial time, problem 1 is *NP*. Obviously, the complexity of checking whether the entry  $w$  is matched with (or covered by) an entry  $y_i$  in  $RvNDB$  is  $O(m)$  ( $m$  is the number of attributes). Then the complexity of checking whether  $RvNDB$  covers the entry  $w$  is  $O(m \cdot |RvNDB|)$  ( $|RvNDB|$  is the number of entries in  $RvNDB$ ). Therefore, checking whether an entry  $w$  is the solution of problem 1 can be done in  $O(m \cdot |RvNDB|)$ , and problem 1 is *NP*.

**Lemma 2.** Any CNF-SAT instance  $\phi$  can be converted to a real-valued negative database  $RvNDB_\phi$ .

*Proof.* Giving any CNF-SAT instance  $\phi$  with  $n$  clauses and  $m$  variables  $x_1, x_2, \dots, x_m$ ,  $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_n$ , a real-valued negative database  $RvNDB_\phi = \{y_1, y_2, \dots, y_n\}$  with  $m$  attributes and  $n$  entries can be constructed as follows.

- (1) Divide the domain of each attribute into two segments:  $[l_k, p_k)$  and  $[p_k, u_k]$  ( $k = 1, 2, \dots, m$ ). Then encode three intervals as follows.

$$\begin{cases} [l_k, p_k) = 0 \\ [p_k, u_k] = 1 \\ I_k = * \end{cases} \quad (2)$$

As the interval  $I_k$  covers both interval  $[l_k, p_k)$  and  $[p_k, u_k]$ , the symbol '\*' represents either 0 or 1.

- (2) Each clause  $C_i$  is mapped to an entry  $y_i$  of  $RvNDB_\phi$ , and a binary negative database denoted as  $eNDB_\phi = \{e_1, e_2, \dots, e_n\}$  is constructed according to  $RvNDB_\phi$ .
  - (a) If the  $k^{\text{th}}$  variable is presented as  $x_k$  in  $C_i$ ,  $[l_k, p_k)$  is assigned to  $y_i[k]$ , and  $e_i[k]$  is set as 0.
  - (b) If the  $k^{\text{th}}$  variable is presented as  $\bar{x}_k$  in  $C_i$ ,  $[p_k, u_k]$  is assigned to  $y_i[k]$ , and  $e_i[k]$  is set as 1.
  - (c) If the  $k^{\text{th}}$  variable does not appear in  $C_i$ ,  $I_k$  is assigned to  $y_i[k]$ , and  $e_i[k]$  is set as '\*'.

After all the clauses of the CNF-SAT instance are mapped to entries, the real-valued negative database  $RvNDB_\phi$  denoted with intervals is constructed. The database  $eNDB_\phi$  is the binary form of  $RvNDB_\phi$  and they can be converted to each other easily. The  $eNDB_\phi$  has the same structure with the binary negative database defined in (Esponda et al., 2004b; Esponda et al., 2009).

**Lemma 3.** Any entry in  $eNDB_\phi$  is not the true assignment of  $\phi$ .

*Proof.* Obviously, if assign  $e_i[k]$  ( $k = 1 \dots m$ ) to the  $k^{\text{th}}$  variable  $x_k$ , the entry  $e_i$  is not a true assignment of  $C_i$ . Because each entry in  $eNDB_\phi$  cannot satisfy at least one clause of  $\phi$ , it is not the true assignments of  $\phi$ .

**Lemma 4.** Each true assignment of the CNF-SAT instance  $\phi$  corresponds to a real-valued entry not covered by  $RvNDB_\phi$  and vice versa.

*Proof.* For any true assignment  $a$  of the CNF-SAT instance  $\phi$ , as every clause of  $\phi$  is satisfied by  $a$  and every entry in  $eNDB_\phi$  is not the true assignment of  $\phi$ , at least one bit of  $a$  is different from each entry in  $eNDB_\phi$ . That is to say,  $a$  is not covered by  $eNDB_\phi$ . According to equation 2, the assignment  $a$  can be converted to an entry  $v$  that consists of intervals, and obviously the entry  $v$  is not covered by  $RvNDB_\phi$ .

For any entry  $w$  consists of  $m$  real values and not covered by any entries in  $RvNDB_\phi$ , it could be encoded to a binary string  $a$  as follows.

$$a[k] = \begin{cases} 0 & w[k] \in [l_k, p_k) \\ 1 & w[k] \in [p_k, u_k] \end{cases}, \quad k = 1, 2, \dots, m \quad (3)$$

As  $w$  is not covered by any entry  $y_i$  ( $i = 1, \dots, n$ ) in  $RvNDB_\phi$ , there is at least one attribute  $k$  that  $w[k]$  is not covered by  $y_i[k]$ , and the encoding result  $a[k]$  is different from  $e_i[k]$  as well, i.e.  $a[k] = \bar{e}_i[k]$ . According to the encoding of  $RvNDB_\phi$ , if assign  $a[k]$  to  $x_k$ , the clause  $C_i$  will be satisfied. Moreover, since  $w$  is not covered by all the entries in  $RvNDB_\phi$ , all the clauses in  $\phi$  are satisfied by  $a$ , and  $a$  is a true assignment of the CNF-SAT instance  $\phi$ .

**Theorem 1.** Problem 1 is *NP*-complete.

*Proof.* According to lemma 4, the problem of checking the satisfiability of the instance  $\phi$  is equivalent to the problem 1 for  $RvNDB_\phi$ . Furthermore, due to the instance  $\phi$  is chosen arbitrarily, any instance of the CNF-SAT can be converted to a special real-valued negative database. Therefore, problem 1 is *NP*-complete.

**Theorem 2.** Problem 2 is *NP*-hard.

*Proof.* Based on lemma 1, 2, 3, 4 and theorem 1, this theorem is immediately proved.

## Generation Algorithm for $RvNDB$ s

Some generation algorithms for the binary negative database have been proposed (Esponda et al., 2004b; Jia et al., 2005; Esponda et al., 2007a; Esponda et al., 2009; Liu et al., 2011). Based on these generation algorithms, an algorithm for generating real-valued negative databases is proposed in this section.

Giving a positive database  $DB = \{x_1, x_2, \dots, x_n\}$ , and there are  $m$  attributes in  $DB$ . Each entry in  $DB$  is a vector of  $m$  real values. The procedure of the generation algorithm for the real-valued negative database from  $DB$  is described as follows.



- (1) Preprocessing: Divide the domains of attributes in  $DB$ , and convert  $DB$  to a real-valued database  $DB^I$  which consists of intervals.
- (2) Encoding: Encode  $DB^I$  to a binary positive database  $DB^2$ .
- (3) Generating: Input  $DB^2$  to an algorithm for generating a binary negative database from the binary positive database such as the  $q$ -hidden algorithm or the prefix algorithm, and output a binary negative database  $NDB^2$ .
- (4) Decoding: Decode  $NDB^2$  to a real-valued negative database  $RvNDB$  which consists of intervals.

### Phase 1: Preprocessing

The preprocessing phase contains two processes: the dividing process and the converting process. In the dividing process, the domain of each attribute in  $DB$  is divided into several distinct intervals. In the converting process, the values of each entry in  $DB$  are converted to the intervals which they belong to.

**Dividing Process.** The domain of each attribute in  $DB$  is divided to a set of intervals. For any  $k$  ( $k = 1 \dots m$ ), the interval set  $P_k = \{p_{k,1}, p_{k,2}, \dots, p_{k,num_k}\}$ , where  $num_k$  is the number of intervals in  $P_k$ .

The set  $P_k$  should be generated according to the requirements for real-life applications and satisfy following basic conditions.

- (1) The union of all the intervals in  $P_k$  equals to  $I_k$ , i.e.

$$p_{k,1} \cup p_{k,2} \cup \dots \cup p_{k,num_k} = I_k \quad (4)$$

- (2) The intersection between any two different intervals in  $P_k$  is the empty set, i.e.

$$p_{k,i} \cap p_{k,j} = \emptyset, \quad \forall 1 \leq i < j \leq num_k \quad (5)$$

- (3) Since  $DB$  will be encoded to a binary database, ideally, the number of intervals in  $P_k$  should be the exponent of 2.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Divide algorithm</b><br>Input: $I = \{I_1, I_2, \dots, I_m\}$ , $Num = \{num_1, num_2, \dots, num_m\}$<br>Output: $P = \{P_1, P_2, \dots, P_m\}$                                                                                                                                                                                                                                                                                                                                                                                                      |
| <ol style="list-style-type: none"> <li>1. <b>For</b> the <math>k^{th}</math> (<math>k = 1 \dots m</math>) attribute <b>do</b></li> <li>2.   <math>low \leftarrow l_k</math>, <math>unit \leftarrow (u_k - l_k)/num_k</math></li> <li>3.   <b>For</b> <math>i = 1</math> to <math>num_k - 1</math> <b>do</b></li> <li>4.     <math>up \leftarrow low + unit</math></li> <li>5.     Add <math>[low, up)</math> to <math>P_k</math></li> <li>6.     <math>low \leftarrow up</math></li> <li>7.   Add <math>[low, u_k]</math> to <math>P_k</math></li> </ol> |

**Figure 1.** An algorithm for dividing process

Although the dividing process depends on the requirements of real-life applications, a simple algorithm is given in figure 1. The algorithm in figure 1 equally divides each domain  $I_k$  ( $k = 1 \dots m$ ) into  $num_k$  intervals. This algorithm can be applied to

some applications such as the privacy-preserving data publication.

**Converting Process.** According to above dividing process,  $DB$  can be converted to a real-valued positive database  $DB^I$  which consists of intervals as follows.

Let  $DB^I = \{t_1, t_2, \dots, t_n\}$ . For each entry  $x_i$  ( $i = 1 \dots n$ ) in  $DB$ , the value of the  $k^{th}$  ( $k = 1 \dots m$ ) attribute is converted to the interval which  $x_i[k]$  belongs to in  $P_k$ , i.e.

$$t_i[k] = p_{k,j} \quad \text{iff} \quad x_i[k] \in p_{k,j} \quad (6)$$

### Phase 2: Encoding

In order to generate real-valued negative databases, the real-valued positive database  $DB^I$  is encoded to a binary database, and then an algorithm for generating negative databases from binary positive databases can be employed.

For the  $k^{th}$  ( $k = 1 \dots m$ ) attribute, since any two different intervals are not intersected with each other and the number of the intervals in  $P_k$  is the exponent of 2, it is easy to encode  $num_k$  intervals in  $P_k$  as  $num_k$  binary strings with length  $\log_2(num_k)$ . According to the encoding of intervals in  $P_k$ , the entries in  $DB^I$  can be converted to binary strings. The details of the encoding phase are shown in figure 2.

The algorithm shown in Figure 3 is used for generating the binary code from an integer. If the length of the binary code is less than  $l$ , some zeros will be attached after it. It follows that all the generated binary strings have the same length. In the encoding phase, this algorithm is employed to encode the intervals in  $P_k$  ( $k = 1 \dots m$ ) according to their indexes.

#### Encode algorithm

Input:  $DB^I = \{t_1, t_2, \dots, t_n\}$ ,  $P = \{P_1, P_2, \dots, P_m\}$   
 Output:  $DB^2 = \{s_1, s_2, \dots, s_n\}$

1. **For** each entry  $t_i$  ( $i = 1 \dots n$ ) in  $DB^I$  **do**
2.   **For** the  $k^{th}$  ( $k = 1 \dots m$ ) attribute **do**
3.      $l \leftarrow \log_2(num_k)$
4.     **If**  $t_i[k]$  is the  $j^{th}$  interval in  $P_k$  **then**
5.        $s_i[k] \leftarrow \text{binaryCode}(j-1, l)$

**Figure 2.** The algorithm for encoding phase

#### binaryCode(v, l)

Input: an integer  $v$  and length  $l$   
 Output: a binary string  $str$  with length  $l$

1.  $i \leftarrow 1$
2. **While**  $i \leq l$  **do**
3.    $str[i] \leftarrow v \bmod 2$
4.    $v \leftarrow v/2$
5.    $i \leftarrow i + 1$

**Figure 3.** Generating the binary code from an integer

### Phase 3: Generating

In the encoding phase, a binary database  $DB^2$  has been generated from the real-valued positive database  $DB^1$ . In the generating phase,  $DB^2$  is inputted to an algorithm for generating negative databases from the binary positive database, such as the prefix algorithm (Esponda et al., 2004b; Esponda et al., 2009), the *RNDB* algorithm (Esponda et al., 2004b; Esponda et al., 2009) and the *q*-hidden algorithm (Jia et al., 2005; Esponda et al., 2007a), and the generation algorithm outputs a binary negative database  $NDB^2 = \{z_1, z_2, \dots, z_N\}$ .

### Phase 4: Decoding

In the generating phase, a binary negative database  $NDB^2$  is obtained from the binary positive database  $DB^2$ . It is not convenient to use the binary negative database in the real-valued space. Therefore, in the decoding phase, the binary negative database  $NDB^2$  is converted to a real-valued negative database *RvNDB*.

|                                                                            |  |
|----------------------------------------------------------------------------|--|
| <b>Decode algorithm</b>                                                    |  |
| Input: $NDB^2 = \{z_1, z_2, \dots, z_N\}$ , $P = \{P_1, P_2, \dots, P_m\}$ |  |
| Output: <i>RvNDB</i>                                                       |  |
| 1. <b>For</b> each entry $z_i$ ( $i = 1 \dots N$ ) in $NDB^2$ <b>do</b>    |  |
| 2. <b>For</b> the $k^{\text{th}}$ ( $k = 1 \dots m$ ) attribute <b>do</b>  |  |
| 3. $Q_k \leftarrow \text{Extend\_Pattern}(z_i[k], P_k)$                    |  |
| 4. $RvNDB \leftarrow RvNDB \cup (Q_1 \times \dots \times Q_m)^{**}$        |  |

**Figure 4.** The algorithm for decoding phase

|                                                                                                                     |  |
|---------------------------------------------------------------------------------------------------------------------|--|
| <b>Extend_Pattern(<i>str</i>, <math>P_k</math>)</b>                                                                 |  |
| Input: A string <i>str</i> defined upon alphabet $\{0, 1, *\}$ , and $P_k$                                          |  |
| Output: A set <i>W</i> of intervals which is decoded from <i>str</i>                                                |  |
| 1. Initialize <i>W</i> as the empty set                                                                             |  |
| 2. Set $B_p$ as the unspecified positions of <i>str</i>                                                             |  |
| 3. <b>For</b> every possible assignment <i>T</i> of $B_p$ <b>do</b>                                                 |  |
| 4.     Let <i>str'</i> be the same with <i>str</i> but the unspecified positions are assigned according to <i>T</i> |  |
| 5.     Let <i>temp</i> be the decimal value of <i>str'</i>                                                          |  |
| 6.     Add $p_{k, \text{temp}+1}$ to <i>W</i>                                                                       |  |
| 7. Merge the adjacent intervals in <i>W</i>                                                                         |  |

**Figure 5.** Decoding a string defined upon alphabet  $\{0, 1, *\}$  to a set of intervals

The algorithm for the decoding phase is given in figure 4. Since the entries in  $NDB^2$  are defined upon the alphabet  $\{0, 1, *\}$ , and the symbol ‘\*’ represents either 0 or 1 at a given position, each entry may cover multiple strings of specified

\*\* It is noted that the cross product operation  $(Q_1 \times \dots \times Q_m)$  could be compressed to a new type of entry  $y_Q = (Q_1, Q_2, \dots, Q_m)$  for decreasing the size of *RvNDB*. Consequently, an entry in *RvNDB* could consist of *m* sets of intervals.

values (i.e. 0 and 1). An extra algorithm for decoding a string defined upon the alphabet  $\{0, 1, *\}$  to a set of intervals is given in figure 5.

The algorithm in figure 5 enumerates every specified string which is covered by the string *str*, and converts these specified strings to intervals. Finally, the adjacent intervals in *W* are merged.

## Application to the Privacy-Preserving Data Publication

As sensitive data has been involved in many applications nowadays, the privacy preserving of data has been widely concerned. The privacy-preserving data publication is a technique which can both preserve the privacy and maintain the utility of the published data.

The data generalization is an important technique for protecting sensitive data and preserving privacy (Fung et al., 2010). In the preprocessing phase of the generation algorithm for the real-valued negative database, the conversion from real values to intervals can be regarded as the generalization of real values, and the dividing of domains determines the generalized intervals. Therefore, when apply the real-valued negative database to the privacy-preserving data publication, the first phase can be replaced by some generalization techniques, such as some algorithms that can satisfy the *k*-anonymity principle (Sweeney, 2002). Then, a real-valued negative database can be generated from the generalized positive database through the generation algorithm described in the former section.

An example of applying the real-valued negative database to the privacy-preserving data publication is given as follows. The original data is shown in table 2. There are four attributes in the original positive database, and the attribute ‘Name’ is the explicit identifier. The combination of attributes <Age, Postcode> is regarded as the quasi-identifiers. The sensitive attribute is ‘Salary’. The domains of the last three attributes (i.e.  $I(\text{Age})$ ,  $I(\text{Postcode})$  and  $I(\text{Salary})$ ) are divided as follows.

$I(\text{Age}) = [0, 150]$

Divided as:  $\{[0, 20), [20, 40), [40, 70), [70, 150]\}$

Encoded as:  $\{00, 01, 10, 11\}$ .

$I(\text{Postcode}) = [00000, 50000]$

Divided as:  $\{[00000, 10000), [10000, 20000), [20000, 30000), [30000, 50000]\}$

Encoded as:  $\{00, 01, 10, 11\}$ .

$I(\text{Salary}) = [0, 100.0]$

Divided as:  $\{[0, 10.0), [10.0, 20.0), [20.0, 50.0), [50.0, 100.0]\}$

Encoded as:  $\{00, 01, 10, 11\}$ .

**Table 2.** The original database

| Name  | Age | Postcode | Salary (k\$) |
|-------|-----|----------|--------------|
| Alice | 16  | 21000    | 5.5          |
| Bob   | 10  | 25000    | 65.0         |
| John  | 55  | 16000    | 50.5         |
| Bill  | 62  | 11000    | 25.3         |
| David | 42  | 13000    | 15.5         |

The generalized data which satisfies 2-anonymity principle (the  $k$ -anonymity principle demands that each entry in the published database cannot be distinguished from at least other  $k-1$  entries (Sweeney, 2002)) is shown in table 3. The binary positive database is shown in table 4. The binary negative database generated by the prefix algorithm (Esponda et al., 2004b; Esponda et al., 2009) from the binary positive database is shown in table 5. Finally, the real-valued negative database decoded from the binary negative database is shown in table 6.

**Table 3.** The real-valued positive database which consists of intervals

| Age      | Postcode       | Salary (k\$)  |
|----------|----------------|---------------|
| [0, 20)  | [20000, 30000) | [0, 10.0)     |
| [0, 20)  | [20000, 30000) | [50.0, 100.0] |
| [40, 70) | [10000, 20000) | [50.0, 100.0] |
| [40, 70) | [10000, 20000) | [20.0, 50.0)  |
| [40, 70) | [10000, 20000) | [10.0, 20.0)  |

**Table 4.** The binary positive database

| Age | Postcode | Salary (k\$) |
|-----|----------|--------------|
| 00  | 10       | 00           |
| 00  | 10       | 11           |
| 10  | 01       | 11           |
| 10  | 01       | 10           |
| 10  | 01       | 01           |

**Table 5.** A binary negative database

| Age | Postcode | Salary (k\$) |
|-----|----------|--------------|
| 01  | **       | **           |
| 00  | 0*       | **           |
| 00  | 11       | **           |
| 00  | 10       | 01           |
| 00  | 10       | 10           |
| 11  | **       | **           |
| 10  | 1*       | **           |
| 10  | 00       | **           |
| 10  | 01       | 00           |

**Table 6.** The real-valued negative database

| Age       | Postcode       | Salary (k\$) |
|-----------|----------------|--------------|
| [20, 40)  | [00000, 50000] | [0, 100.0]   |
| [0, 20)   | [00000, 20000) | [0, 100.0]   |
| [0, 20)   | [30000, 50000] | [0, 100.0]   |
| [0, 20)   | [20000, 30000) | [10.0, 20.0) |
| [0, 20)   | [20000, 30000) | [20.0, 50.0) |
| [70, 150] | [00000, 50000] | [0, 100.0]   |
| [40, 70)  | [20000, 50000] | [0, 100.0]   |
| [40, 70)  | [00000, 10000) | [0, 100.0]   |
| [40, 70)  | [10000, 20000) | [0, 10.0)    |

## Discussion

The real-valued negative database can be applied to the privacy-preserving data publication. The preprocessing phase of the generation algorithm for the real-valued negative database could be replaced by an existing generalization algorithm. The privacy of the published data is preserved through not only the generalization but also the real-valued negative database. If high data precision is expected, the generalized intervals can be controlled to small ranges. Even if the sensitive data is not generalized, it is still under the protection of the negative representation. If the real-valued negative database is complete, it can be considered as “equivalent” to the generalized positive database and no extra information is lost. Furthermore, since the relationship between the real-valued positive database and the real-valued negative database is one-to-many, and it is hard to check whether two real-valued negative databases correspond to the same positive database (the hardness could be roughly controlled through the generation algorithm for negative databases). Therefore, the real-valued negative database could be properly applied to the privacy-preserving data republication (Xiao and Tao, 2007) and the privacy-preserving publication of dynamic data (Jian et al., 2007; Xiao and Tao, 2007; Bu et al., 2008).

## Conclusions and Future Work

Since the data in some applications is naturally represented in real-valued space, it is difficult to apply binary negative databases properly. Therefore, the real-valued negative database is proposed in this paper. Reversing the real-valued negative database is proved to be an  $NP$ -hard problem, and it follows that the real-valued negative database could be employed to protect data privacy. Based on the generation algorithms for the binary negative database, an effective algorithm for generating real-valued negative databases is proposed in this paper.

The real-valued negative database is applied to the privacy-preserving data publication in this paper. The privacy of the published data is under the protection of both the generalization and the negative representation. Furthermore, the balance between security and data precision could be controlled through the level of generalization and the generation algorithm for the real-valued negative database.

Although the definition and a generation algorithm for the real-valued negative database are given in this paper, some further work is expected. Since the generation algorithm for the real-valued negative database is based on the generation algorithms for the binary negative database, some more efficient generation algorithms which are dedicated to the real-valued negative database are expected to be proposed. Some operations for the real-valued negative database such as select, delete, insert, project, union, intersection, set difference, Cartesian product and join need to be designed urgently. These database operations are critical for extending the applications of the real-valued negative database. Moreover, some concrete and practical solutions of applying the real-valued negative database to the privacy-preserving data publication will be considered in future as well.

## Acknowledgements

This work is partly supported by National Natural Science Foundation of China (No. 61175045).

## References

- Bu, Y., Fu, A. W. C., Wong, R. C. W., Chen, L. and Li, J. (2008). Privacy Preserving Serial Data Publishing by Role Composition. *Proceedings of the VLDB Endowment*, 1(1): 845-856.
- Dasgupta, D. and Azeem, R. (2007). A Negative Authentication System. Technical Report, Department of Computer Science, The University of Memphis: CS-07-001.
- Dasgupta, D. and Azeem, R. (2008). An Investigation of Negative Authentication Systems. *Proceedings of the Third International Conference on i-Warfare & Security (ICIW 2008)*, Omaha, Nebraska, USA.
- Esponda, F. (2006). Negative Surveys. *Arxiv: math/0608176*.
- Esponda, F., Ackley, E. S., Forrest, S. and Helman, P. (2004a). Online Negative Databases. *Proceedings of the Third International Conference on Artificial Immune Systems*, Catania, Sicily, Italy.
- Esponda, F., Ackley, E. S., Forrest, S. and Helman, P. (2005). Online Negative Databases (with Experimental Result). *International Journal of Unconventional Computing*, 1(3): 201-220.
- Esponda, F., Ackley, E. S., Helman, P., Jia, H. and Forrest, S. (2007a). Protecting Data Privacy through Hard-to-Reverse Negative Databases. *International Journal of Information Security*, 6(6): 403-415.
- Esponda, F., Forrest, S. and Helman, P. (2004b). Enhancing Privacy through Negative Representations of Data. Technical Report, Department of Computer Science, University of New Mexico: A667894.
- Esponda, F., Forrest, S. and Helman, P. (2009). Negative Representations of Information. *International Journal of Information Security*, 8(5): 331-345.
- Esponda, F., Trias, E. D., Ackley, E. S. and Forrest, S. (2007b). A Relational Algebra for Negative Databases. Technical Report, Department of Computer Science, University of New Mexico: TR-CS-2007-2018.
- Forrest, S., Perelson, A. S., Allen, L. and Cherukuri, R. (1994). Self-Nonself Discrimination in a Computer. *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, IEEE Computer Society.
- Fung, B. C. M., Wang, K., Chen, R. and Yu, P. S. (2010). Privacy-Preserving Data Publishing: A Survey of Recent Developments. *ACM Computing Surveys*, 42(4): 1-53.
- González, F., Dasgupta, D. and Niño, L. (2003). A Randomized Real-Valued Negative Selection Algorithm. *Proceedings of the 2nd International Conference on Artificial Immune Systems (ICARIS 2003)*, Springer Berlin Heidelberg.
- Horey, J., Groat, M., Forrest, S. and Esponda, F. (2007). Anonymous Data Collection in Sensor Networks. *Proceedings of the Fourth Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, Philadelphia, USA.
- Ji, Z. and Dasgupta, D. (2004). Real-Valued Negative Selection Algorithm with Variable-Sized Detectors. *Proceedings of the 2004 Conference on Genetic and Evolutionary Computation (GECCO 2004)*, Springer Berlin Heidelberg.
- Ji, Z. and Dasgupta, D. (2006). Applicability Issues of the Real-Valued Negative Selection Algorithms. *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation (GECCO 2005)*, ACM, Seattle, Washington, USA.
- Ji, Z. and Dasgupta, D. (2007). Revisiting Negative Selection Algorithms. *Evolutionary Computation*, 15(2): 223-251.
- Jia, H., Moore, C. and Strain, D. (2005). Generating Hard Satisfiable Formulas by Hiding Solutions Deceptively. *Proceedings of the Twentieth National Conference on Artificial Intelligence*, AAAI Press, Pittsburgh, Pennsylvania.
- Jian, P., Jian, X., Zhibin, W., Wei, W. and Ke, W. (2007). Maintaining k-Anonymity against Incremental Updates. *Proceedings of the 19th International Conference on Scientific and Statistical Database Management*.
- Liu, R., Luo, W. and Wang, X. (2011). A Hybrid of the Prefix Algorithm and the q-hidden Algorithm for Generating Single Negative Databases. *Proceedings of the 2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS 2011)*, IEEE Computer Society, Paris, France.
- Selman, B., Kautz, H. and Cohen, B. (1995). Local Search Strategies for Satisfiability Testing. *Proceedings of the Second DIMACS Challenge on Cliques, Coloring, and Satisfiability*.
- Sweeney, L. (2002). k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5): 557-570.
- Xiao, X. and Tao, Y. (2007). m-Invariance: Towards Privacy Preserving Re-publication of Dynamic Datasets. *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, ACM, Beijing, China.



## On the Role of the AIS Practitioner

Emma Hart<sup>1</sup>, Mark Read<sup>2</sup>, Chris McEwan<sup>3</sup>, Uwe Aickelin<sup>4</sup> and Julie Greensmith<sup>4</sup>

<sup>1</sup>Edinburgh Napier University

<sup>2</sup>University of York

<sup>3</sup>Microsoft Research

<sup>4</sup>University of Nottingham

e.hart@napier.ac.uk

### Abstract

Cognisant of the gulf between engineers and immunologists that currently hinders a truly inter-disciplinary approach to the field of Artificial Immune Systems (AIS), we propose a redefinition of the term *AIS practitioner*, as an individual who identifies those components and interactions captured in computational immunology models that are responsible for a particular property of interest (POI), and distils from these a set of algorithms and principles that can be applied in an engineering domain. We outline the role of the cross-disciplinary practitioner and the potential benefits to the field.

### Introduction

The Artificial Immune Systems (AIS) field, which seeks to understand immune system operation and exploit these principles in engineering contexts, has been criticized for reasoning by metaphor (Stepney et al., 2005): algorithms are based on naïve biological analogies, are poorly understood, and weakly capture their inspiring biological properties. Various frameworks have been proposed that enable the richness of emergent biological properties to be translated into useful engineered systems. The conceptual framework (Stepney et al., 2005) proposed starting with a study of the immunological system, modeling it, and leading to the development of engineering algorithms. (Andrews, 2008) notes that the framework lacks guidance on selecting biological inspiration for particular engineering domains, and that evaluating a particular domains potential requires that it first be modeled. The immuno-engineering framework (Timmis et al., 2008), through better grounded in engineering by accounting for the physical properties of engineering systems (e.g. in terms of memory, processing power etc.), suffers the same problem.

The interdisciplinary approach to algorithmic development advocated by these frameworks is essential but raises practical issues: immunologists inform modeling efforts to realistically represent and aid biological understanding; engineers desire algorithms that can be theoretically analyzed, verified and validated to demonstrate desirable properties

and applicability to the problem at hand - the biological inspiration is irrelevant. Furthermore, engineers and immunologists will rarely interact; they have differing goals.

We propose redefining the term AIS practitioner, as an individual who identifies those components and interactions captured in computational immunology models that are responsible for a particular property of interest (POI), and to distil from these a set of algorithms and principles that can be applied in an engineering domain.

### The AIS Practitioner

The cross-disciplinary AIS practitioner possesses sufficient understanding to identify key modern-day engineering challenges, and appreciate how qualities exhibited by the immune system can help. They can comprehend computational immunology models, and derive therefrom the required algorithmic principles. Stripping away immunological nomenclature is vital; algorithmic components labeled T cells or lymph nodes are so abstract at this stage that they are meaningless and confusing.

We capture the AIS practitioners role as shown in figure 1. The box represents the space of abstract representations (models, simulations, algorithms, design principles) that capture a particular POI, a quality observed in the immune system and desired in an engineering solution, such as robustness, homeostasis, life-long learning, and adaptation, to name some examples (Hart and Timmis, 2008). Abstract representation complexity is represented vertically, whereas the domain of concern is depicted horizontally.

The bottom right corner represents the minimal rational representation (MRR), the minimal set of components/actors, interactions or principles responsible for manifestation of the POI, expressed free of immunological terminology. Its minimalism reflects the computational power and memory constraints of many engineering domains. The top right corner represents inefficient, sub-optimal and/or obfuscated abstract representations. Although they capture the POI, they contain superfluous interactions or components, and as such are not considered as suitable for adoption in engineering contexts. The top left contains detailed and com-

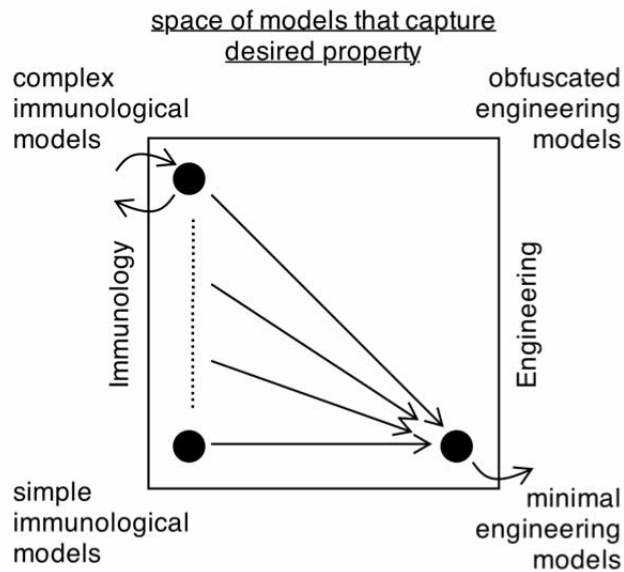


Figure 1: The space of abstract representation that capture a property of the immune system

plex models, representing, for example, large numbers of cells, pathways and spatial compartments. They may capture a great many immunological properties, in addition to the POI. The bottom left corner contains the minimal set of components and interactions responsible for the POI, and no other properties. Though minimal, these abstract representations exist in a form to benefit immunology, not in a form immediately applicable in engineering domains.

We consider the AIS practitioners role to derive the MRR, given some abstract representation occupying the left of the box. The components and interactions present that are redundant or unnecessary with respect to the POI are removed or abstracted. Depending on the nature of the POI, and the immune-inspired engineering solution being derived, the MRR may be expressed as an algorithm, or as design principles for constructing systems. The level of abstraction of the model is such that it should specify all necessary components of the system, and the interactions that take place between components. The model should be able to be validated to show that it gives rise to the desired POI, and if applicable, under what parameter ranges.

## Conclusion

We have outlined a role for the AIS practitioner in defining design principles that capture particular properties of the immune system. Such principles are free from immunological jargon and thus can easily be interpreted by engineers wishing to solve a problem. We believe this will increase the usage of immune-inspired solutions to engineering problems, through facilitating access to the subject by those unfamiliar

with the biological field, and secondly, by providing principles that are readily understood in terms of their computational properties.

## References

- Andrews, P. (2008). *An investigation of a methodology for the development of artificial immune systems: a case-study in immune receptor degeneracy*. PhD thesis, University of York.
- Hart, E. and Timmis, J. (2008). Application areas of ais: the past, present and future. *Journal of Applied Soft Computing*, 8.
- Stepney, S., Smith, R., Timmis, J., Tyrrell, A., Neal, M., and Hone, A. (2005). Conceptual frameworks for artificial immune system. *International Journal of Unconventional Computing*, 1.
- Timmis, J., Hart, E., Hone, A., Nea, M., Robins, A., Stepney, S., and Tyrrell, A. (2008). Immuno-engineering. In *2nd IFIP International Conference on Biologically Inspired Collaborative Computing*, pages 3–17. IEEE Press.

## Accelerating Immunos 99

Paul Taylor<sup>1,2</sup> and Fiona A. C. Polack<sup>2</sup> and Jon Timmis<sup>2,3</sup>

<sup>1</sup> BT Research & Innovation, Adastral Park, Martlesham Heath, UK, IP5 3RE

<sup>2</sup> Department of Computer Science, University of York, UK, YO10 5DD

<sup>3</sup> Department of Electronics, University of York, UK, YO10 5DD  
paul.n.taylor@bt.com, {fiona.polack,jon.timmis}@york.ac.uk

### Abstract

Immunos 99 is a classification algorithm based upon the principles of Artificial Immune Systems (AIS). AIS algorithms can provide alternatives to classical techniques such as decision trees for classification tasks. Immunos 99 provides one alternative however the algorithm and implementations have some room for performance improvement. This paper discusses improvements made to Immunos 99 and its reference implementation to improve runtime performance. The new algorithm/implementation results in an approximate 40% reduction in run time on test data. The paper closes with a proposal for an implementation of the Immunos 99 algorithm that is suitable for use on map/reduce clusters.

### Introduction

The classification of large data sets is increasingly important in a wide variety of applications. Many classification techniques are available, with varying applicability. The need to apply classification techniques to large datasets has motivated research on improving the efficiency or running time of algorithms. For example the AIRS classifier (Watkins, Timmis, & Boggess, 2004) has been successfully parallelised, reducing execution time while retaining classification performance (Watkins & Timmis, 2004).

Our interest in the classification of large data sets relates to the automation of business process analysis. A business process is a set of activities that are required in a specific order, to provide a product or service. Efficient organisation of business processes is essential to the efficient running of the business itself. The data collected by monitoring of business processes is not clean, which limits the range of applicable classification techniques (Taylor, Leida, & Majeed, 2012). Techniques which are both resistant to noise in training data and applicable across a wide variety of input data are therefore of particular interest in this endeavour.

In initial small-scale experimentation on samples of poor quality data from real business processes, the Immunos 99 algorithm, proposed by Brownlee (Brownlee, 2005), showed promising performance. However the runtime performance of this (and other) classifier made its use infeasible for large data sets. Immunos 99 is a classification algorithm

inspired by the Immunos family, which incorporates additional immune-inspired techniques such as cell proliferation and hypermutation. A complete specification for Immunos 99 can be found in (Brownlee, 2005, Section 5). The only publicly-available implementation of the algorithm is a plugin for the Weka data mining toolkit (Hall et al., 2009; Brownlee, n.d.).

### Motivation & Experimental Environment

We are working on monitoring data for business processes relating to provision of service and customer interactions. The ability to predict and prevent the failure of business processes in these areas would enable organisations to perform more efficiently, achieving improved levels of customer service and customer satisfaction. There is little research on the automated analysis of such business processes, not least because of the poor quality of data available. However, if classification is to be useful industrially, it has to be achieved fast enough to allow timely intervention in failing business processes.

In small-scale trials, we tested a range of classifiers, seeking effective prediction of the outcome of a business process is considered; since the business process outcome can be only *SUCCESS* or *FAILURE*, based on some business condition, the problem is a two-class classification task. We find that Immunos 99 can be trained to handle classification of the real data, which is noisy and of low quality.

We next trained Immunos 99 with a typical dataset from BT process monitoring. The proprietary dataset comprises observations on 65,000 business processes, over 46 variables; 74% of the business processes were identified as belonging to the class *SUCCESS*, and the remaining 26% were classified as *FAILURE*. The trial performed 10-fold cross-validation on the dataset, for each classifier under test. However, the execution of the training runs was infeasibly long, with typical executions taking around 40 hours to complete classification of a day's test data.

Since Immunos 99 was the most successful classification technique in small-scale trials, we decided to improve the implementation of the algorithm, exploiting multi-

threading, with the goal of reducing execution time. What constitutes an appropriate execution time in this work is dependent on business context – the execution must complete in time to be able to intervene in a business process that may take hours, or may take weeks to complete. Our initial goal was to reduce execution time so that training on one day’s data takes hours rather than days.

The purpose and goal of our work on the Immunos 99 algorithm is to enable training and classification of large, noisy data sets, at an appropriate speed for the business context. We do *not* include rigorous algorithmic analysis, as this is not appropriate for our business context.

The improvements discussed here have been tested across large proprietary datasets, and the improvement in execution time has been shown throughout. Results from one of the proprietary datasets is included, to show improvements in execution time are achievable on the real datasets. However, owing to the proprietary nature of the data, the paper illustrates performance by application to a public dataset (Section ), allowing detailed discussion and analysis, as well as presentation of reproducible results.

### Analysing the Original Immunos 99

Immunos 99 is an immune-inspired classifier originally described by Brownlee (Brownlee, 2005) (Algorithm 1). It utilises a number of immune system features, and analogues of immune system components in its design. In an immune system, an antigen is something that provokes a response from the immune system; in this case, an antigen is analogous to an observation from the input data. In an immune system, the B-cells are recognisers that bind to specific antigens with a particular affinity; here the analogy is of recognisers with an affinity represented as a numerical value.

Here, we use complexity functions to reason about relative execution times, independent of implementation variations. Assuming that each operation in the function takes 1 unit of time, the estimated runtime for a sequential execution of the algorithm is equal to its complexity. Changes to the defined functions to account for the expected reduction in runtime caused by the application of multi-threading and parallelism are discussed in the following section. The complexity functions are derived from examination of the original implementation of the algorithm in (Brownlee, n.d.).

To calculate the complexity function, let  $g$  represent the number of generations to use and  $n$  represent the number of observations in the input data, which is also the size of the initial B-cell pool. The algorithm performs  $g * n^2$  affinity calculations. After each B-cell has been exposed and the affinity calculated, fitness is determined by the B-cell’s rank position in the B-cell pool. To determine the rank position requires a sort operation, of complexity in the order of  $O(n \log(n))$ .

To train the algorithm,  $n$  is the size of the training set. There are  $c$  classification classes, and a total of  $a$  antibodies

---

**Algorithm 1:** Original Immunos 99 Training Algorithm, from (Brownlee, 2005)

---

```

1 Divide data into antigen groups (by classification label)
2 foreach group do
3   Create initial B-Cell population
4   foreach generation (to parameter limit) do
5     Expose all B-cells to all antigens from all groups
6     Calculate fitness scoring
7     Perform population pruning
8     Perform affinity maturation
9     Insert randomly selected antigens of the same group
10  end
11 end
12 Perform final pruning for each B-cell population
13 return the final B-cell population as the classifier

```

---

ies in pool, where  $a_c$  is the number of antibodies for class  $c$ . The approximate complexity of the training phase of Immunos 99 is formed of the three parts: the initial generation of antibodies in equation 1 (line 3); the generation and mutation phase in equation 2 (lines 5-9); and the final pruning phase in 3 (line 12). The complexity of the overall training implementation is the sum of the three phases – the function shows a typical quadratic relationship, plotted against size.

$$2n + a + c \quad (1)$$

$$g(c(a_c + n(7a_c + 2(a_c \log(a_c)))))) \quad (2)$$

$$c(a_c + n(3a_c + a_c \log(a_c))) \quad (3)$$

### Parallelising the Algorithm

A number of changes have been made, to improve training times for the Immunos 99 algorithm. These improvements fall into 2 categories, first partial threading of the algorithm itself, and second changes to the Java implementation (published in (Brownlee, n.d.)). Partial threading is introduced using the Fork-Join paradigm which is supported by the core Java library in Java 7.

The first opportunity for parallelism, training of each group of B-cells in parallel, is identified by Brownlee (Brownlee, 2005), but not included in the Weka implementation.

The most time consuming part of the algorithm is the affinity calculations across all antibodies and all antigens. Each affinity calculation (the affinity of one antigen to one B-Cell) is independent of all the other affinity calculations. This is a significant opportunity for further parallelism to be introduced. The nested loops running the affinity calculations in series (lines 5-6 of Algorithm 1) can be replaced with a construct that distributes the individual calculations



to the available thread pool, and then collects the results as each thread completes. Once all calculations are complete, the normal sequence of the algorithm is resumed.

An example of the fork-join construct is shown in figure 1. The modified algorithm is summarised in Algorithm 2. The fork-join construct is used for the blocks between lines 7-13 and lines 22-30, with the worker threads performing lines 10 and 25 only.

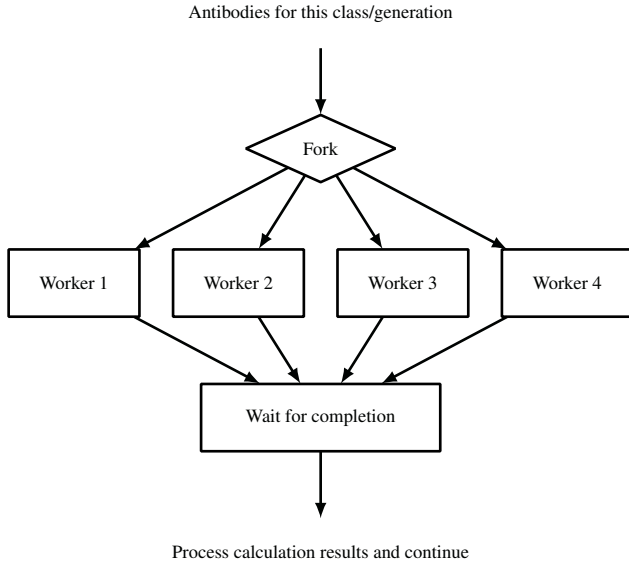


Figure 1: Example fork-Join construct for antibody affinity calculation

The parallelised algorithm uses the same number of operations to train the classifier as the original Immunos 99 algorithm, so the complexity function is the same. If we assume that each computation takes a single time unit, the expected runtime of the modified algorithm is reduced, since affinity calculations occur simultaneously. For the three phases of the algorithm described for the original, and a thread pool of size  $t$ , the estimated runtime of the initial setup phase is unchanged (equation 1). The estimated runtime of the modified generation and mutation phase, and of the modified final pruning phase are given by equations 5 and 4, respectively. The runtime of the algorithm is primarily influenced by the number of classes in the training data (since each group of antibodies can be trained in parallel with enough threads) and by the number of execution threads available (reducing the time required to complete the affinity calculations by allowing more to be performed in parallel).

$$g\left(\frac{a_c + n\left(\frac{a_c}{t} + (6a_c) + 2(a_c \log(a_c))\right)}{\min(c, t)}\right) \quad (4)$$

$$c(a_c + n(2(\frac{a_c}{t}) + a_c + a_c \log(a_c))) \quad (5)$$

Figure 2 shows the effect of additional threads on the complexity of the algorithm for a hypothetical 4 class prob-

---

#### Algorithm 2: Parallelised Immunos 99 algorithm

---

```

1 Create the thread pool,  $p$ 
2 Divide data into antigen groups (by classification label)
3 foreach group do
4   Create initial B-Cell population
5   foreach generation (to parameter limit) do
6     begin Expose all B-cells to all antigens from all groups
7       foreach Antigen,  $ag$ , in the training set do
8         foreach Antibody,  $ab$  in the generation do
9           begin on a thread from  $p$ 
10            Calculate the affinity between  $ag$  and  $ab$ 
11          end
12        end
13      end
14    end
15    Calculate fitness scoring
16    Perform population pruning
17    Perform affinity maturation
18    Insert randomly selected antigens of the same group
19  end
20 end
21 begin final pruning for each B-cell population
22 foreach Antigen,  $ag$ , in the training set do
23   foreach Antibody,  $ab$  in the last generation do
24     begin on a thread from  $p$ 
25       Calculate the affinity between  $ag$  and  $ab$ 
26     end
27   end
28    $ab_{best} = ab$  with best affinity to  $ag$ 
29   increment score of  $ab_{best}$ 
30 end
31 Prune antibodies with score below threshold
32 end
33 return the final B-cell population as the classifier

```

---

lem – using more than  $c$  threads has only a small additional benefit. Figure 3 illustrates the effect of increasing the number of classes, where the thread pool size is equal to the number of classes.

#### Other Parallelisation: the AIRS Approach

Watkins, Timmis, and Boggess report parallelisation of the AIRS algorithm (Watkins, Timmis, & Boggess, 2004; Watkins & Timmis, 2004), which we look to for inspiration. In the AIRS parallelisation the training data is partitioned and sent to threads which train independently before merging the resulting memory cells into a single pool for classification. The AIRS approach is more pervasively parallel than

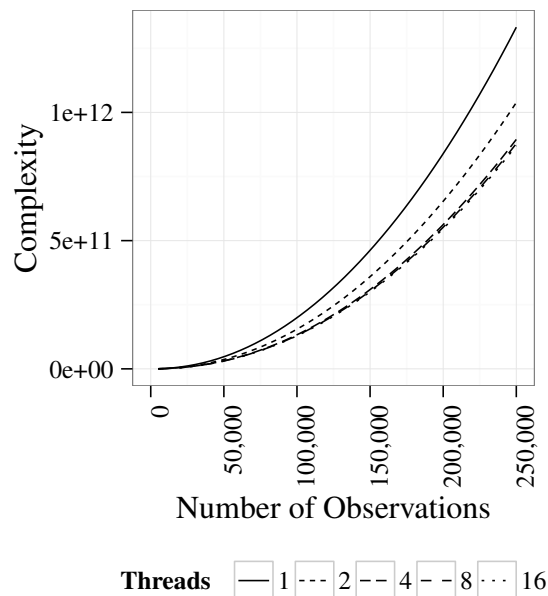


Figure 2: Effect of additional threads on the estimated run-time of modified Immunos 99 on 4 class data.

the Immunos 99 parallelisation, as it distributes large, intensive, independent units of work. In contrast, the Immunos 99 parallelisation approach attempts to gain the largest increase in performance possible without extensive redesign of the original code – the modified Immunos 99 implementation only minimally changes the serial version to distribute the large number of computationally simple affinity calculations.

### Reducing Implementation Inefficiencies

The original implementation of the Immunos 99 algorithm is part of the Weka Classification Algorithms project on SourceForge (Brownlee, n.d.). We analysed the execution of the original Weka code using the VisualVM profiler (Oracle Corporation, n.d.). VisualVM identifies the hot spots in the code where improvements are most likely to have an impact. The profiler analysis was used both to guide the algorithmic changes in the previous section and, in conjunction with code reviews, to identify inefficiencies in the use of Java and its libraries. These implementation-specific improvements affect the runtime of the Immunos 99 modified algorithm, but are not factored into the general complexity analysis in Sections and , as they pertain to specific versions of the implementation, Java library and virtual machine.

A significant time overhead in the Weka implementation is the use of array copy operations. For instance, each time the algorithm calls `Collections.sort` from the Java library sorting functions, the collection is copied into a temporary array, sorted, and copied back, rewriting the original collection. We reduce the time to sort a collection by

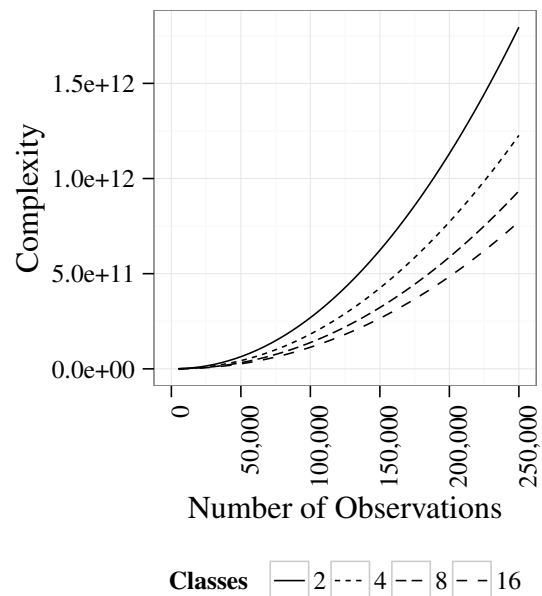


Figure 3: Impact of number of classes on estimated runtime of modified Immunos 99 (with  $t = c$ ).

implementing a sort function that directly modifies the array behind each collection without any copying. A simpler modification to reduce copying is achieved by refactoring the Weka code to avoid unnecessary use of methods that rely on copying, such as `Instance.toDoubleArray()`.

The profiler analysis showed extensive use of `LinkedList` objects. `LinkedLists` are inefficient when used in conjunction with random access operations, as in the original code, since every index request requires traversal of the list (average complexity  $O(n/2)$ ). To improve efficiency, the `LinkedList` objects were replaced with `ArrayList` objects, which provide the same interface but have complexity of  $O(1)$  access to contents.

A review of the logic of the Weka implementation revealed an inefficiency in the final pruning step (Algorithm 1 line 7). At this step, only the antibody with the best affinity is required, so it is unnecessary to sort all the antibodies by affinity. By replacing a call to `Collections.sort` with a call to `Collections.min`, the complexity is reduced from  $O(n \log(n))$  (for sorting) to  $O(n)$  (to scan the collection).

### Experimental Results

We illustrate the difference in execution time between the original and modified implementations using the *Chess (King-Rook vs. King-Knight) Data Set* (kr-kk) from the UCI machine learning repository (Frank & Asuncion, 2010). Each test is a 10-fold cross validation of the kr-kk dataset, which is repeated 10 times, with the time for each run recorded. Tests are performed on an otherwise-idle Win-

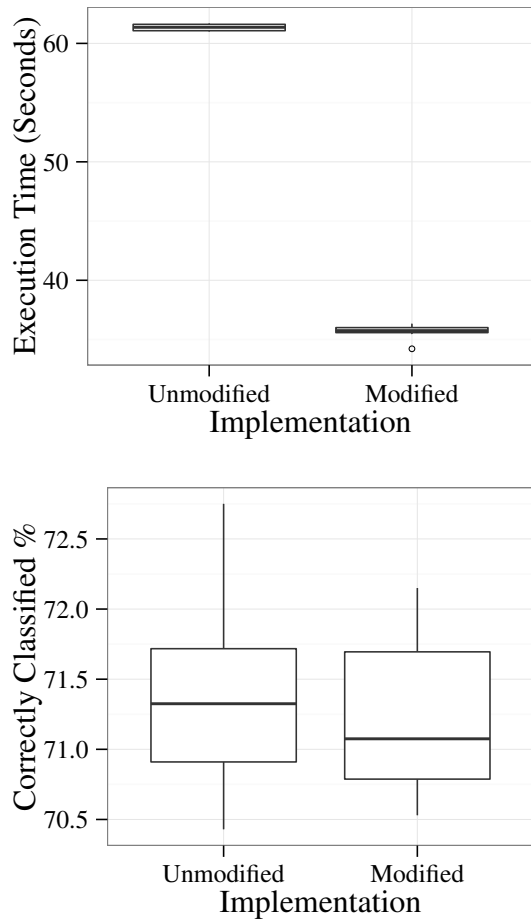


Figure 4: Box plots showing execution times and accuracy for 10-fold cross-validation of each implementation on the chess data set

dows Server 2008 box with 16 Cores (4x Intel Xeon E7340 CPUs) and 32GB of RAM. Tests are run under the 64-bit Java 7 JDK (version 1.7.0-b147).

Figure 4 shows the time and classification performance of the implementations. The box plots show the median value as the centre bar; the outer limits of the box are the first and third quartiles, and whiskers show the highest and lowest value within  $1.5 * InterquartileRange$  of the box edge. Any outliers are shown as points.

The results for execution time (upper plot in Figure 4) shows that the improved algorithm has a mean execution time that is approximately 58% of the execution time for the original. Both implementations exhibit very low levels of variation in execution time. The outlier in the plot for the modified algorithm is thought to be due to the influence of the operating system scheduler on the server performing the tests.

The lack of overlap between the result ranges of the two

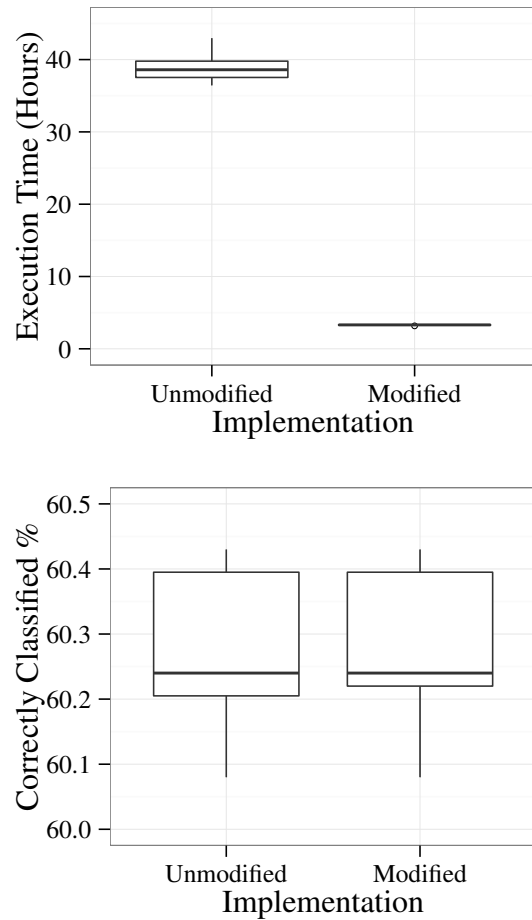


Figure 5: Box plots showing execution times and accuracy for 10-fold cross-validation of each implementation on the proprietary data set

implementations indicates a significant improvement in execution time with the modified implementation. A non-parametric Mann-Whitney U test for significance was performed, with  $H_0$  of no significant difference between the performance of the original and modified implementations. The test p-value is 0.0001806, so  $H_0$  is rejected with greater than 99% confidence.

To increase confidence that the classification ability of the algorithm has not been compromised by the modifications, the overall classification accuracy was also recorded for the 10 runs (lower plot in Figure 4). In the results, the modified algorithm has a slightly lower median accuracy, but less variation in classification performance than in the original implementation. Performing the Mann-Whitney U test for significance, with  $H_0$  that there is no significant difference between the performance of each algorithm yields a p-value of 0.6305 so  $H_0$  is accepted.

In the motivation for this work (Section ), we describe

the training phase execution of the original algorithm on a proprietary dataset, which took in the order of 40 hours to complete. Performance of the modified algorithm was also tested on this proprietary dataset. The experimental setup was identical to that for the kr-kk dataset. The mean execution time was reduced from 38.86744 hours for the unmodified algorithm to 3.29218 hours for the modified algorithm. A Mann Whitney U Test for significance was again performed, with  $H_0$  that the execution time for the two implementations is the same. The p-value of 0.00001083 determines that  $H_0$  is rejected with greater than 99% confidence. The mean accuracy of the algorithms is almost identical: 60.274% and 60.276% for the unmodified and modified implementations, respectively (Mann Whitney U Test p-value of 0.9092). The results (following the same format as before) are plotted in figure 5.

The results are in line with expectation: the quadratic complexity of the Immunos 99 algorithm means that performance gains from the modified implementation should be higher for larger datasets. We have also informally observed increased improvements for larger datasets in other work on proprietary datasets.

### Further Work: Proposed Map-Reduce Implementation

In working on the Immunos 99 algorithm, it became apparent that a map-reduce variant of Immunos 99 would support massively-parallel execution. This should lead to feasible training times for very large datasets. The outline is presented here as an opportunity for further work.

Map-reduce (Dean & Ghemawat, 2008; The Apache Software Foundation, n.d.) allows tasks to be broken down into an arrangement of *map* and *reduce* tasks. Map tasks are responsible for splitting the incoming data into chunks that can be independently processed. Reduce tasks are responsible for processing a chunk of data and then returning a result to the next task in the chain. The most time consuming part of the Immunos 99 algorithm (exposing all B-cells to all antigens) could leverage these constructs as follows.

1. Scoring of each antibody within a generation can be performed independently of any other antibody.
2. Each antigen/antibody affinity comparison used to calculate the antibody score can be mapped out as a separate task.
3. Ranking the antibodies by affinity for scoring could be a subsequent reduce task that merges the antibody scores into a sorted list for the calculation (cf. merge sort).
4. Rank scoring of the antibodies for each instance can be done as a reduce task on the sorted antibody scores.
5. Final pruning and other finalising tasks can be performed in serial by a final reduce task.

The final pruning step could be implemented using a similar series of map and reduce tasks, which would further reduce execution time.

The map-reduce approach would allow a refactoring that can efficiently exploit a large number of compute resources to process in a much shorter time than either the original implementation in Section , or the improved variant in Section . A caveat to this is that the time spent administering and distributing the map and reduce tasks over a compute cluster is not insignificant, so for smaller datasets the single machine approach from Section is likely to be faster.

### Conclusion

This paper has assessed the practical performance of the Immunos 99 classification algorithm (Brownlee, 2005), considering some performance deficiencies of the algorithm and its publicly-available implementation (Brownlee, n.d.). We present modifications to the existing source code that enable Immunos 99 training to run in an acceptable time (according to the business context) on large, noisy datasets, by effective exploitation of multi-core processors are discussed. We outline additional modifications using the map-reduce paradigm (Dean & Ghemawat, 2008; The Apache Software Foundation, n.d.) which have the potential to support massively-parallel execution.

### References

- Watkins, A., Timmis, J., & Boggess, L. C. (2004). Artificial Immune Recognition System (AIRS) : An Immune Inspired Supervised Machine Learning Algorithm. *Genetic Programming and Evolvable Machines*, 5(3), 291–317.
- Watkins, A. & Timmis, J. (2004). Exploiting parallelism inherent in AIRS, an artificial immune classifier. In G. Nicosia, V. Cutello, P. J. Bentley, & J. Timmis (Eds.), *Artificial immune systems* (Vol. 3239, pp. 427–438). Lecture Notes in Computer Science. Springer.
- Taylor, P., Leida, M., & Majeed, B. (2012). Case study in process mining in a multinational enterprise. In K. Aberer, E. Damiani, & T. Dillon (Eds.), *Data-driven process discovery and analysis* (Vol. 116, pp. 134–153). Lecture Notes in Business Information Processing. Springer Berlin Heidelberg.
- Brownlee, J. (2005). *Immunos-81, The Misunderstood Artificial Immune System* (Technical report No. 1-02, Swinburne University of Technology, Australia).
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1), 10–18. doi:10.1145/1656274.1656278
- Brownlee, J. (n.d.). WEKA Classification Algorithms. Retrieved from <http://weka.classalgos.sf.net/>
- Oracle Corporation. (n.d.). VisualVM. Retrieved from <http://visualvm.java.net/>
- Frank, A. & Asuncion, A. (2010). UCI machine learning repository. Retrieved from <http://archive.ics.uci.edu/ml>
- Dean, J. & Ghemawat, S. (2008, January). MapReduce. *Communications of the ACM*, 51(1), 107. doi:10.1145/1327452.1327492
- The Apache Software Foundation. (n.d.). Apache Hadoop. Retrieved from <http://hadoop.apache.org/>



# Modelling the Migration and Maturation of Dendritic Cells for Automatic Optimization of Complex Engineering Problems

N.M.Y. Lee<sup>1</sup> and H.Y.K. Lau<sup>2</sup>

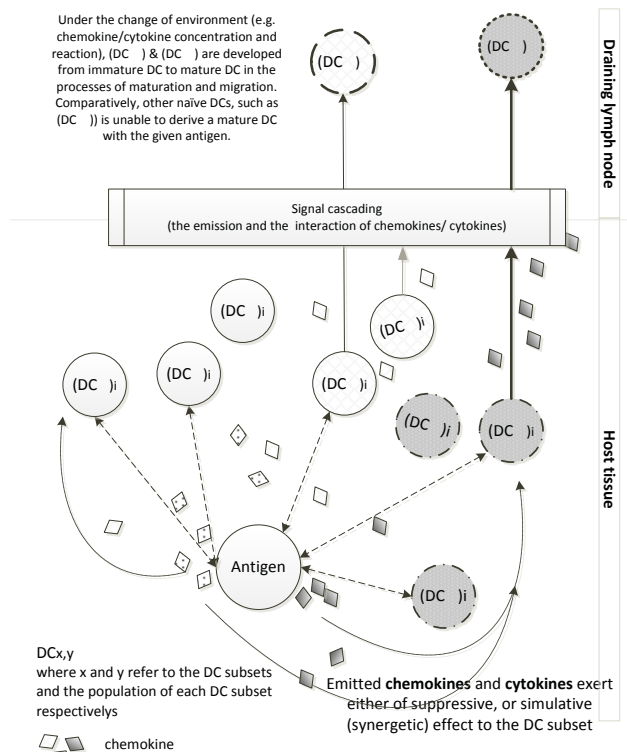
<sup>1,2</sup> The University of Hong Kong  
[myleenicole@gradaute.hku.hk](mailto:myleenicole@gradaute.hku.hk), [hyklau@hku.hk](mailto:hyklau@hku.hk)

## Abstract

The metaphor of Migration and Maturation of Dendritic Cells (DCs), in particular, the induced signal pathways to the change of the immunogenic functions of DCs (Martin-Fontecha, Lanzavecchis & Sallusto, 2009) provides important features for the development of the proposed DC-inspired optimization algorithm. Specifically, the quantified capability and behavioral change of DCs (Callard, George & Stark, 1999) of the classical DC models (Caetano Reis, 2006), namely, (i) ontogeny of DC, (ii) the selectively up-regulation of markers (of DC subsets), (iii) the level of threat (of the antigen), (iv) production of chemokine and cytokine, (v) transcription factors, and (vi) the effector functions of DCs underpin a highly autonomous control mechanism for the evolution of the optimal solution(s).

In the proposed autonomous optimization framework, a multi-agent system is developed as described in (Lee & Lau, 2012) incorporating the following DC-inspired philosophies,

- Compartmental interaction and communication of agents.
- Potential threat of solutions is scaled as DCs perform in the host tissue, as well as the measurement of the fitness of the solution(s).
- Synergetic signal cascading (i.e. cytokine production) (Ricart et al., 2011) is adopted to facilitate the solution development process with self-governing characteristic.
- Effector functions are defined to gauge the quality of solution during the solution development (for example, to recruit more DC subsets and re-generating more candidate solutions as depicted in Fig.1).



**Fig. 1.** Schematic diagram of the interactions between DC subsets and chemokine in maturation and migration where DCs are represented by (DC<sub>x,y</sub>)<sub>i</sub> and (DC<sub>x,y</sub>)<sub>m</sub>

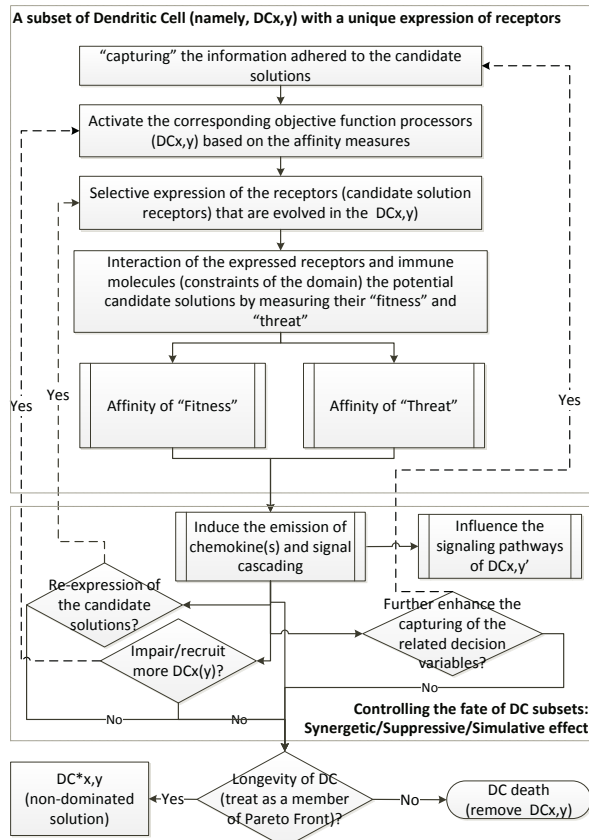
In the proposed framework (as depicted in Fig. 2), each artificial DC uptakes the characteristics of an antigen (or a permutation) for assessing the level of “fitness” and “threat” of the solution, which are critical to the signal production and forthcoming pathways. Similar to the classical optimization algorithm, “fitness” refers to the optimality for the given objective functions. Whilst the “threat” introduced in the proposed DC-inspired optimization algorithm is quantified by the virulence factors (to the domains), number of iterations or number of replications. These quantified “threat” perhaps ruining the quality of solutions instead of the measurement of “fitness”.

Further enhancing the autonomous control of the proposed framework, immunological signal cascading is adopted. Biologically, signal cascading is represented by a highly dynamic and complex network of immunological signals, namely, chemokines and cytokines. They are predominantly emitted based on the quantified “threat” and “fitness” as abovementioned. The emitted signals and their interactions will further stimulate/suppress the production of chemokine and cytokine. For each of produced signals, it has a specific role in governing the behaviors of the artificial DC subsets, particularly aims at presenting the best for the activation of T-cell. With the inspirations of these immuno-features, a metaphor is anticipated for evolving the optimal solution(s) in the optimization problem. Primarily, the following immunological effector functions are implemented in the proposed DC-based optimization framework,

- Ontology – e.g. recruiting new populations (of artificial DC subsets).
- Phenotype – e.g. changing the permutations (re-capturing the characteristics of the given problem).

- Population – e.g. the death rate of artificial DC subsets, and the rate of proliferation.
- Migratory behavior.

By modeling DCs' capabilities and behaviors changes in maturation and migration, namely, the quantified level of "threat" and a self-regulated metaphor, the optimal solutions will be resulted. More importantly, the solutions generation can be specific and avoid premature converge which is observed in the classical optimization algorithms.



**Fig. 2.** Proposed automatic DC-mediated Cascading Framework for pursuing an optimal solution – For the attack, each DC subset (denoted by DC<sub>x,y</sub>) acts as a “decision maker” (or “controller”) is equipped with a distinct phenotype for gauging the threat of the candidate solutions (as antigens) based on (i) the emitted signals (as cytokines and chemokine), and (ii) their interactions with the corresponding receptors, and other immune molecules from the neighbouring DC subsets.

In parallel to the studies of classic DC maturation models, impact of the properties changes of “ontogeny” (Population subsets), “phenotype” and “effector” to solutions development are studied in the proposed DC-inspired optimization framework. More importantly, the study reveals some of the unexplored immuno-phenomenon and mechanisms through Matlab simulation.

## References

- Martin-Fontecha, A., Lanzavecchia, A. & Sallusto, F., Dendritic Cell Migration to Peripheral Lymph Nodes, In: Lombardi, G. & Riffio-Vasquez (eds.) Dendritic Cells, pp.31- 49, Springer, Heidelberg (2009)
- Caetano Reis e Sousa: Dendritic cells in mature age, Nature Review Immunology, June 2006, pp.476-473 (2006)
- Callard, R., George, A. J., Stark, J.: Cytokines, chaos, and complexity. Immunity 11, 507–513 (1999)
- Lee, N.M.Y., Lau, H.Y.K.: A Cooperative Multi-objective Optimization Framework based on Dendritic Cells Migration Dynamics, AI-2012, pp.201-206 (2012)
- Ricart, B.G., John, G., Lee, D., Hunter, C.A. & Hammer, D.A., Dendritic Cells Distinguish Individual Chemokine Signals through CCR7 and CXCR4, J Immunol, 186, pp.53-61(2011)

# A hierarchical support vector machine based on feature-driven method for speech emotion recognition

Lingli Yu<sup>1</sup>, Binglu Wu<sup>1</sup> and Tao Gong<sup>2,3,4</sup>

<sup>1</sup> School of Information Science and Engineering, Central South University, Changsha, 410083, P.R. China

<sup>2</sup> College of Information S. & T., Donghua University, Shanghai 201620, China

<sup>3</sup> Engineering Research Center of Digitized Textile & Fashion Technology, Ministry of Education, Donghua University, Shanghai 201620, China

<sup>4</sup> Department of Computer Science, Purdue University, West Lafayette 47907, USA

llyu@csu.edu.cn   taogong@dhu.edu.cn

## Abstract

Through the analysis of one-vs.-one, one-vs.-rest and the decision tree mechanism of binary support vector machine emotion classifiers, a method based on feature-driven hierarchical support vector machine is proposed for speech emotion recognition. For each layer, classifier used different feature parameters to drive its performance, and each emotion is subdivided layer by layer. This method did not rely entirely on the activity-valence dimensional emotion model, but relied on the type of emotion to distinguish. Furthermore, classifications are constructed by appropriate characteristic parameters ultimately. Experiments on the Chinese-speaker-dependent and Berlin-speaker-independent corpus reached conclusions as follows, Chinese-speaker-dependent recognition rate is relatively higher than Berlin-speaker-independent. feature-driven hierarchical support vector machine in the case driven by effective features improves the speech emotion recognition performance. Meanwhile applying the mean of the log-spectrum to this method can identify high-activity and low-activity emotion effectively.

Keywords: Feature-driven, Speech emotion recognition, Support vector machine (SVM), Hierarchical Classifier, Mean of the Log-Spectrum (MLS)

## 1. Introduction

Speech emotion recognition is not only used for human-computer interaction, but also applied in speech synthesis, artificial counseling, polygraph, telephone banking, driverless system and so on[1]. Nowadays, the field of emotion recognition is facing to huge challenges. The main difficulty [2] is that we could not extract a most effective universal phonetic feature for various kinds of emotion. Furthermore, one sentence may contain several kinds of emotions at the same time, and emotions may be associated with just parts of a sentence. There is no clear boundary for each complex emotion. Sometimes even humans are not capable of distinguishing them. Moreover, their cultural backgrounds and the environments also affect the emotion expression. Speech emotion recognition is a hot research issue in natural computing area, some researchers and institutions have done many works for emotion recognition [3].

Emotion recognition system consists of three parts: the module for extracting feature parameters, the module for reducing feature parameters' dimensions and the module for emotion recognition. Most researchers mainly utilized prosody features [4-6] like pitch period, short time energy, duration of voice, and their relative statistics as feature parameters. Besides, MFCC (Mel Frequency Cepstrum Coefficient) was also used for emotion recognition. This coefficient has more information when it is extracted from voiced sound rather than from unvoiced sound. Yang[7] proposed a coordinate feature set based on music theory for emotion recognition. They considered that the acoustic and semantic features were useful for the recognition. Another problem in speech emotion recognition is how to reduce the dimensions of the features in order to simplify the calculation. There were several common ways to do it: Sequential Floating Forward Selection (SFFS), Forward Feature Selection (FFS), Backward Feature Selection (BFS), Principal Component Analysis (PCA) and Linear Discriminate Analysis (LDA). After the simplified features obtained, we needed to build the effective classifiers. In 1990s, most of the emotional models were built on Maximum Likelihood Bayes (MLB) and Linear Discriminant Classification (LDC). Now, there are much more kinds of emotional models being used for emotion classification, like Hidden Markov Model (HMM), Gaussian Mixture Models (GMM), Artificial Neural Network (ANN), Support Vector Machine (SVM) etc. [8], and sometimes we use multiple methods simultaneously. However, we cannot know which classifier is the best at the current time.

SVM stemmed from statistical learning theory proposed by Vapnik and others, as a classifier. SVM could yield good results even from small test samples. So it was widely used for speech emotional recognition. Because of the Structural Risk Minimization, SVM classifier usually had better performance than others [9]. In this paper, an improved method based on feature driven is proposed in order to perfect speech emotional recognition performance. Feature-driven hierarchical SVM does not completely depend on the emotion dimensional model of activation-valence, and it adjusts the feature parameters of each layer gradually according to the property recognized by lots of experiments.

This paper is organized as follows. In section 2, several speech corpora are described, and then Chinese Speaker-dependent and Berlin speaker-independent Speech Corpus are used in this paper. In section 3, three kinds of binary SVM emotion classifications are discussed for a multi-category classification problem of Speech emotion recognition. In section 4, Hierarchical SVM emotion recognition method based on feature-driven is proposed to improve the performance of Speech emotion recognition. We extract six feature parameter classes based on prosody affective features and acoustic affective features to achieve the hierarchical SVM classifiers analysis. Experimental comparison between 4 kinds of hierarchical methods and some analysis for improved the parameters in speech feature driven method are discussed, and compared in section 5. Finally, conclusions are summarized in section 6. The page limit is 8 pages for a full paper. Your submission must be converted to Portable Document Format (PDF). Please be sure to use highest portability and quality options. Papers that significantly deviate from these instructions will not be included.

## 2. Several binary SVM classifications

Support vector machines(SVMs) are one of supervised learning models with associated learning algorithms. Speech emotion recognition of is a multi-category classification problem, here it is converted to binary classification problem to solve one by one, the state-of-the-art including:

### 2.1 One-Versus-One binary SVM

The hyper planes of binary SVM are built from any two of all categories, so the number of binary SVM classifiers is  $k*(k-1)/2$ . Here 'max-wins' voting method is used, for One-Versus-One voting strategy, the  $k*(k-1)/2$  binary SVM classifiers are trained in parallel, For example, category  $i$  and category  $j$  trains with classifier  $C_{ij}$ ,  $C_{ij}$  decides whether sample  $x$  belong to category  $i$  or category  $j$ . Therefore the number of  $i$  category votes adds one, otherwise  $j$ 's number of votes adds one. When the process is over, the category with the most voters is the right category that the sample belongs to. The structure of One-Versus-One binary SVM is shown in figure 1. Here 1-5 are set to represent 5 emotion categories of 2 speech corpora, so 10 classifiers are trained. From the process of category, we know that this method is less effective while the number of the classifiers increases, which will cause the decision speed more slowly.

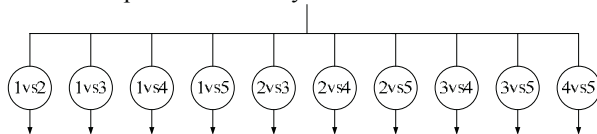


Figure 1: One-Versus-One 'max-wins' voting SVM

### 2.2 One-Versus-The-Rest binary SVM

One-Versus-The-Rest binary SVM only builds  $k$  SVMs, each SVM classifier recognizes one category from all the other categories. The unbalanced decision tree combines the One-Versus-The-Rest with right branch, and it just needs to train  $(k-1)$  classifiers, the number of classifiers is less than them in one-versus-one method. Figure 2 shows the structure of this method, the recognized emotion should be the easiest

to be distinguished on first layer. Chinese Speaker-dependent speech corpus is used, furthermore anger is chosen on the first layer. For instance, 1 represents anger, 2 represents sadness, 3 represents happiness, 4 represents neutral and 5 represents amazement.

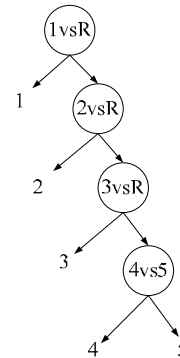


Figure 2: One-Versus-The-Rest unbalanced binary tree SVM

### 2.3 SVM decision Tree Mechanism

The classification error may occurs on any layer of these nodes, and it will spread to all the successor nodes, to this problem, DAG (directed acyclic graph) is proposed by Platt and others [11]. There are  $k*(k-1)/2$  internal nodes,  $k$  branches, each node is a SVM binary classifier. For each test sample, each node's binary decision determines the path of the next decision from the root node. Figure 3 demonstrates the five categories' directed acyclic graph. Whatever the emotion of the test sample is, it will always reach to the bottom of the classifiers[12]. Here, the result is right when every classifier's result is right, but because each binary classifier just handles 2 different emotions, the training is simple and effective. where 1 represents anger, 2 represents sadness, 3 represents happiness, 4 represents neutral and 5 represents amazement in figure 3.

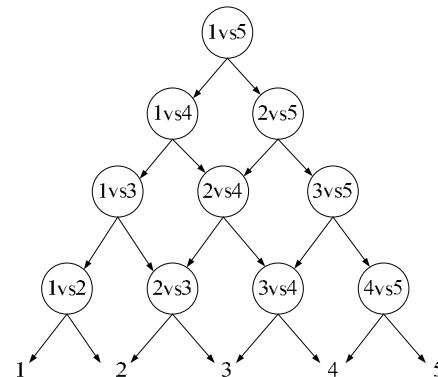


Figure 3: Directed acyclic graph SVM

## 3 Extracting the speech emotion characteristic parameters

Six kinds of feature parameters are extracted to study the hierarchical SVM classifiers based on prosody affective features and acoustic affective features.



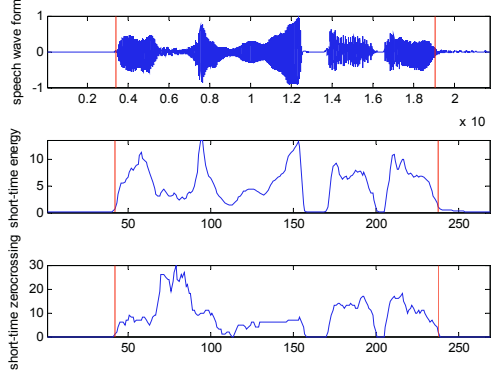
### 3.1 Prosody affective features

Prosody affective feature usually includes intensity, length or duration, pitch, accent, tone, intonation and rhythm.

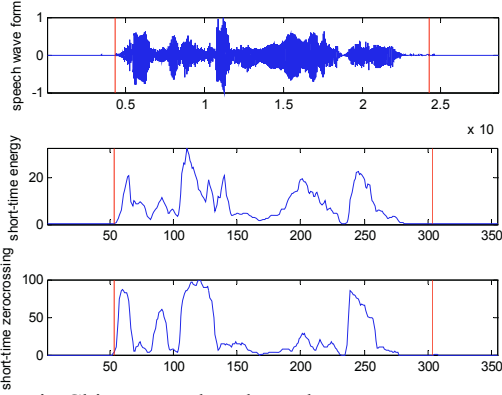
**Short time energy and short time amplitude:**

$$Energy(n) = \sum_{m=0}^{M-1} |S_n(m)|^2, n=0,1,\dots,N-1 \quad (1)$$

Where  $S_n$  is the  $n$ th frame after enframing and windowing,  $N$  is the number of frames,  $M$  is the length of frame. From figure 4, we know that different emotion has different short time energy change, the same is true for short time amplitude.



(a) Sadness in Chinese speaker-dependent



(b) Anger in Chinese speaker-dependent

Figure 4. Voice activity detection using double threshold comparison method

**Short time zero-crossing rate:**

$$Z_n = \sum_{m=-\infty}^{\infty} |\text{sgn}[x(n)] - \text{sgn}[x(n-1)]| \cdot w(n-m) \quad (2)$$

In addition, short time zero-crossing rate combined with short time energy is used in hunting for starting point and end point of the sound, Figure 4 demonstrates this problem, when make final decision, the threshold is needed to set in real situation.

**Pitch period:**

$$\tau_N(k) = \frac{1}{N} \sum_{n=0}^{N-k-1} x(n)x(n+k), 0 \leq k \leq M_o - 1 \quad (3)$$

Where  $t_n$  is the value of autocorrelation function,  $x(n)$  is the recognized voice signal,  $k$  is the delay time,  $M_o$  is the number of the autocorrelation which is calculated in [13]. After getting the autocorrelation function and detecting its peak, we can get pitch period. Figure 5 are shown for Chinese Speaker-dependent speech corpus's pitch period of the same sentence expressed by anger, sadness, neutral and amazement

respectively, and the rate of change in anger and amazement is larger than in sadness and neutral, at the end of the sadness's pitch period envelop curve is cocking up.

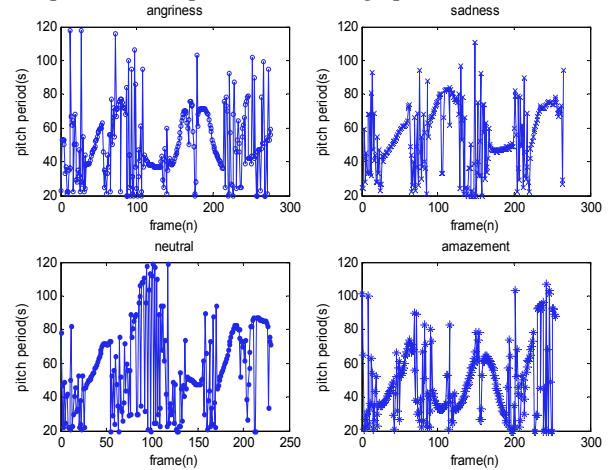


Figure 5. The pitch period of four kind emotions for Chinese speaker-dependent

### 3.2 Acoustic affective features

Acoustic affective features are generally the feature of tone and speech spectrum.

**Formant:**

$$P(w) = 20 \log \left( \frac{1}{|1 + \sum_{k=1}^p a_k e^{-j\omega k}|^2} \right), k=1, 2, \dots, p \quad (4)$$

Levinson-Durbin method is used to calculate the linear prediction coefficient  $a_k()$ . The LPC spectrum are calculated by formula (4), only the first 3 formants are extracted.

**MFCCs:**

The extracting processes of Mel Frequency Cepstral Coefficients (MFCCs) are shown in figure 6.

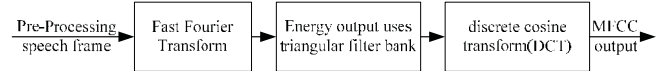
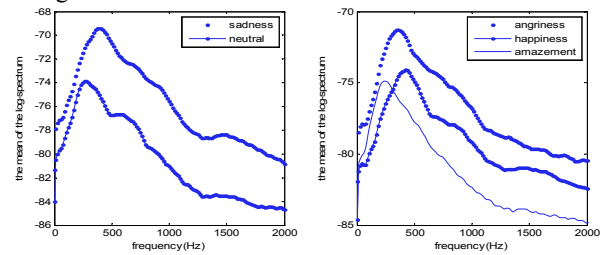


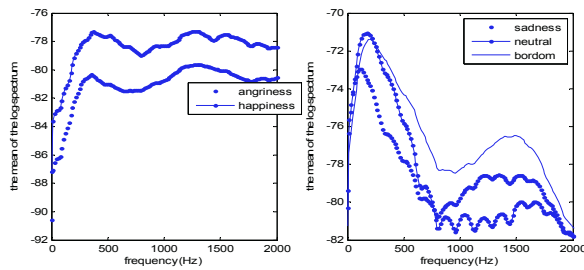
Figure 6: The MFCC extracting process

**Mean of log power spectrum (MLS):**

MLS is calculated by each frame's of one sentence, which is mainly used to convert time domain signal to frequency domain signal. Thus the average of MLS are calculated as showed in formula (5), where  $k$  is spectrum bandwidth,  $N_l$  is the  $l$ th class emotion's utterance number,  $v_d(n, k)$  is the discrete Fourier transformation for the  $n$ th frame of signal  $i$ , the range of  $k$ 's bandwidth is between 0 to 2000Hz.



(a) The average of the log-spectrum for Chinese speaker-dependent



(b) The average of the log-spectrum for Berlin speaker-independent.

Figure 7: The average of the log-spectrum

$$S_i(k) = \frac{1}{N_i} \sum_{i=1}^N \frac{1}{N_i} \sum_{n=1}^N \log |v_{ii}(n, k)| \quad (5)$$

Figure 7 showed the average of the log-spectrum arranged by the similarity of spectrum. (a) is the 5 categories distributed situations of Chinese speaker-dependent. We can see that the MLS peak of anger, happiness concentrated to 400Hz, while the MLS peak of sadness and neutral concentrate to 200 to 250 HZ, MLS envelope curves are similar with each other, especially between high-activity and non-high-activity speech. (b) is the distributed situations of the log power spectrum mean of Berlin speaker-independent. They have similar envelope curves as they are both high-activity emotion. The MLS peak is also concentrated to 400Hz, while sadness, neutral and boredom are concentrated between 100 to 250 Hz. Now, six kinds of parameters and its different dimensions feature derivatives are extracted as showed in table 1.

Table 1: 247 dimensions global feature parameters

| Basic feature parameters        | statistical characteristic                                                                                                  | feature dimens ions |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------|---------------------|
| short time energy and amplitude | Mean, standard deviation, minimum, maximum, dynamic range, mean of first difference, standard deviation of first difference | 14                  |
| short time zero crossing-rate   | Mean, standard deviation, minimum, maximum, dynamic range, mean of first difference, standard deviation of first difference | 7                   |
| pitch period                    | Mean, standard deviation, minimum, maximum, dynamic range, mean of first difference, standard deviation of first difference | 7                   |
| MFCC and its first difference   | Mean, standard deviation, minimum, maximum, dynamic range, mean of first difference, standard deviation of first difference | 168                 |
| First three formants            | Mean, standard deviation, minimum, maximum, dynamic range, mean of first difference, standard deviation of first difference | 21                  |
| 30 dimensions MLS               | mean                                                                                                                        | 30                  |

### 3.3 Reducing dimensions of feature vector

The more dimensions the feature vector has, the more information it contains. However, the calculated complex also greatly increases with the dimensions increases. When the number of vector dimensions exceeds a certain limit, dimension disaster [14] would appear. Therefore, feature selection in broad definition is one kind of mapping transformation from the high-dimensional vector to low-dimensional vector for the sake of reducing dimensions. For literature [15], principal component analysis (PCA) is contributed for reducing dimensions, when PCA is applied to classifiers. it not only reduces calculated quantity, but also eliminates some interference factors. Here, select several characteristic vectors as a main component vectors that correspond to the first  $k$  characteristic value, and  $d$  is the vector dimensions, we set  $k/d$  equals to 0.95, so the number of feature vector's dimensions are reduced from 247 to 31 using PCA in this paper.

### 4. Hierarchical SVM emotion recognition method based on feature-driven

It is shown in Figure 4, A feature-driven hierarchical SVM is proposed for emotion recognition, which demonstrates the structure of this method. In this paper, five kinds of emotions are subdivided for three layers. Especially, Feature-driven hierarchical SVM does not completely depend on the emotion dimensional model of activation-valence, and it adjusts the feature parameters of each layer gradually according to the property recognized by lots of experiments and experience. This feature-driven method is similar with unbalanced decision tree, however, the number of hierarchical layers decreases. A feature-driven hierarchical SVM is strict to each classifier of each layer, generally, we set the two easiest distinguished main categories as the first layer. Therefore, the performance of each classifier of each layer should be well enough to guarantee test samples correctly before it enters the next layer. Meanwhile, linear kernel function is utilized for hierarchical SVM based on feature-driven in this paper.

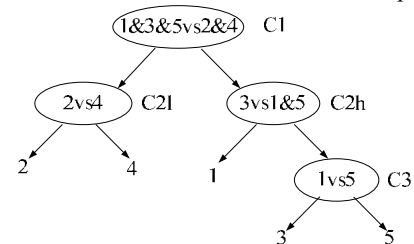


Figure 8: Feature-driven hierarchical SVM

In figure 8, 1 represents anger, 2 represents sadness, 3 represents happiness, 4 represents neutral and 5 represents amazement. The structural parameters are as following for Chinese Speaker-dependent speech corpus. Here, (1) Classifier C1 distinguishes anger, happiness and amazing emotion from sadness, neutral emotion. The feature parameters are the combination of short time energy, short time amplitude, short time zero-crossing rate, pitch period, and MLS (Mean of the Log-Spectrum). (2) Classifier C21 distinguishes sadness from neutral. In addition, the feature parameters of MFCC and formant are the combination.

(3) Classifier C2h distinguishes anger from happiness and amazement. Therefore, parameters includes short time energy, short time amplitude, pitch period, short time zero-crossing rate, mean value of 12 dimensions MFCC and formant combine the feature parameter.

(4) Classifier C3 distinguishes happiness from amazement. The feature parameter is all the feature parameters of C2h combined with MLS.

In the Berlin Speaker-independent speech corpus: 1 represents anger, 2 represents neutral, 3 represents happiness and 4 represents boredom. The structural parameters of Berlin Speaker-independent speech corpus follows. Here

(1) Classifier C1 distinguishes sadness, neutral and boredom from anger and happiness. The feature parameters of short time energy, short time amplitude, pitch period, short time zero-crossing rate and MLS (Mean of the Log-Spectrum) are the combination.

(2) Classifier C21 distinguishes anger from happiness. The feature parameters are the combination with short time energy, short time amplitude, pitch period, short time zero-crossing rate, MLS, and the mean of MFCC.

(3) Classifier C2h distinguishes sadness from neutral and boredom. The feature parameters are the combination of short time energy, short time amplitude, pitch period, short time zero-crossing rate, MFCC and formant.

(4) Classifier C3 distinguishes neutral from boredom, so the feature is the same as C2h.

## 5 Experimental test and result analysis

The orthogonal method is adopted to ensure the independent of each training and test samples. For the experimental process, we choose 50 sentences for each emotion, training samples using 30 of them, test samples using the rest 20. 50 sentences are labeled with 1-50, and every 10 sentences as one group, then there are totally  $C_5^3=10$  situations in the combination of training samples and test samples. Therefore, the 10 independent experiments are conducted to reduce the effect of the unbalance in speech corpus.

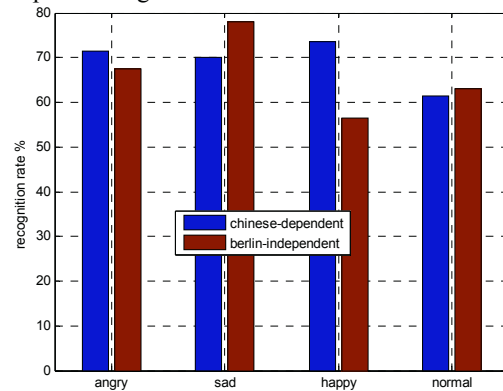
### 5.1 Speech Corpus

Chinese Speaker-dependent and Berlin Speaker-independent Speech Corpus are selected for our experiments. The term of speaker-dependent indicates that all the speech come from one person, which means tones and pronunciation habits are all the same. The Chinese speech corpus records the voice of a woman and includes 5 types of emotions: anger, sadness, happiness, amazement and neutral. Each type of emotion has 50 sentences. The corpus is saved as .wav format with 16 kHz-16 bit resolution. The Berlin corpus [10] records the voice of five man and five women and includes seven types of emotions: happiness, sadness, anger, boredom, disgust, fear and neutral. The corpus is also saved as .wav format with 16 kHz -16 bit resolution.

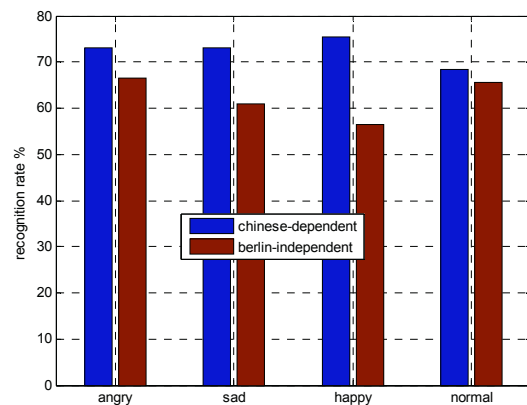
### 5.2 Comparison between Chinese Speaker-dependent and Berlin Speaker-independent speech corpus

We focus on the fallibility between anger, happiness and amazement in Chinese Speaker-dependent speech corpus, and

the fallibility between neutral and sadness. In Berlin Speaker-independent speech corpus, the anger is hard to distinguish from happiness. Meanwhile, the difference between sadness, neutral and boredom are also hard to detect. Therefore, we design the experiments to these two kinds of speech corpus using one versus one method and feature driven method respectively, the results are showed in figure 9. As the amazement in Chinese corpus and boredom in Berlin corpus is not on the same feature space in emotional model, so that just four kinds of emotions (anger, sadness, happiness and neutral) are compared in figure 9.



(a) 1vs1-voting mechanism



(b) feature-driven method

Figure 9: Comparison results of for Chinese and Berlin Corpus

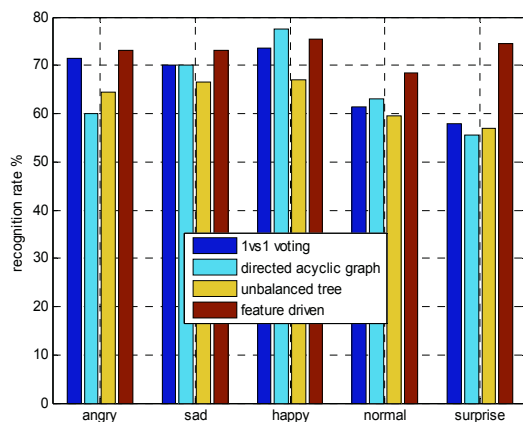
From the results of figure 9, especially for feature driven SVM, we find that the recognition rate for speaker-dependent is obviously higher than for speaker-independent. This is mainly caused by different personal pronunciation habits. Hence, when emotional recognition for speaker-dependent is extended to speaker-independent, the effect of personal pronunciation should be eliminated in feature parameters. As is shown in figure 9, we also know that recognition rate for speaker-independent in one-versus-one mechanism does not decline evidently, which is shown that the 1vs1 recognition method is also available. Combined figure 9 (a) and (b), the emotion recognition rate for speaker-dependent in feature-driven method is almost higher than 1vs1-voting mechanism except for sadness. Meanwhile, the recognition rate for speaker-dependent is also higher than for speaker-independent. Those cause by two reasons. Firstly, there are more feature parameters fused in one-versus-one mechanism each layer which influences the recognition rate. Secondly,

the recognition feature parameters of each layer are more appropriate for Chinese vocal features than German in feature driven SVM.

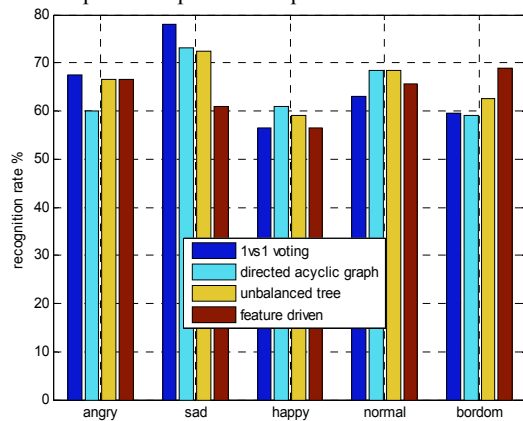
### 5.3 Experimental comparison between four kinds of hierarchical methods

The recognition results of four kinds of hierarchical structure are shown in figure 10. The feature driven method not only keeps the recognition rate for anger, sadness and happiness, but also improves the recognition performance for neutral and amazement. it proves that feature driven method applied to Chinese Speaker-dependent person speech corpus is more effective.

From figure 10, the recognition rate of the one-versus-one mechanism is the highest among all four methods. In addition, recognition rates of unbalanced binary tree and directed acyclic graph are the almost same. The recognition performance of feature driven method is not satisfactory, especially for the sadness and happiness. Therefore, the main reason is that the Chinese pronunciation habit is different from Germany. As we all known that the pause time or silent segments in German is longer than it in Chinese, so the chosen parameters may fit for Chinese but not fit for German.



(a) Chinese Speaker-dependent corpus



(b) Berlin Speaker-independent corpus

Figure 10: Comparison between four kinds of methods

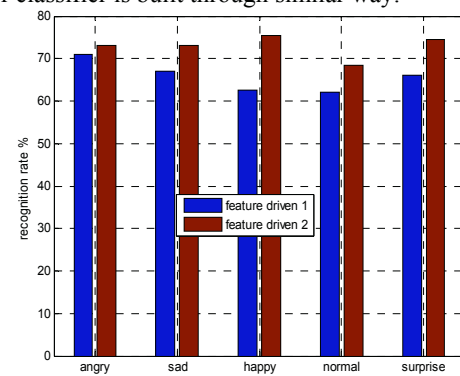
### 5.4 Experiments after improved the parameters in feature driven method of speech emotion recognition

Feature parameters in every layer of hierarchical SVM based on feature-driven are devised and modulated respectively for extracting those feature parameters fitter, meanwhile, for improving the recognition performance. The results after devised and modulated for parameters in feature driven method are showed in table 2.

Table 2: The parameters adjustment and the rate of error identification of feature-driven method

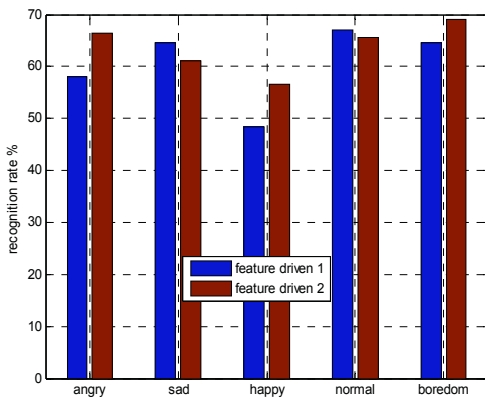
| classifier |                        | Chinese speaker-dependent corpus |                      | Berlin speaker-independent Corpus |                            |
|------------|------------------------|----------------------------------|----------------------|-----------------------------------|----------------------------|
| C1         | feature parameters     | ZEP                              | ZEP + MLS            | ZEP                               | ZEP + MLS                  |
|            | Error recognition rate | 8.7%                             | 6.9%                 | 10.3%                             | 8.1%                       |
| C2l        | feature parameters     | MFCC                             | MFCC + Formant       | MFCC                              | MFCC + ZEP & Formant & MLS |
|            | Error recognition rate | 24.34%                           | 22.25%               | 37.17%                            | 31.09%                     |
| C2h        | feature parameters     | ZEP & MFCC                       | ZEP & MFCC + Formant | ZEP & MFCC                        | ZEP & MFCC + Formant       |
|            | Error recognition rate | 18.57%                           | 16.49%               | 20.61%                            | 21.35%                     |
| C3         | feature parameters     | ZEP                              | ZEP & MFCC + Formant | ZEP                               | ZEP & MFCC + Formant & MLS |
|            | Error recognition rate | 19.34%                           | 12.96%               | 16.24%                            | 15.93%                     |

Here, ZEP represents the combination of short time energy, short time amplitude, pitch period and short zero-crossing rate. Here, given C1 as a instances for Chinese speech corpus to illustrate "Error recognition rate", when high valence emotion (anger, happiness and amazement) is judged to be low valence emotion (neutral, sadness), then it is considered error recognition, and vice versa. A computational method for another classifier is built through similar way.



(a) Chinese Speaker-dependent





(b) Berlin Speaker-independent

Figure 11: Results after improving parameters in Feature-driven methods

Figure 11 is the comparison between before improved and after improved feature driven method. Combining the figure 11 and table 2, we know that the recognition rate is greatly improved after importing the MLS feature parameter. Furthermore, the recognition rate of C21 and C2h classifiers in Chinese Speaker-dependent speech corpus improved after importing the formant, but the recognition rates of the two classifiers are adverse in Berlin Speaker-independent speech corpus. This indicates that a better feature parameters extracted is related to the type of language closely.

## 6. Conclusion

A new feature driven hierarchical SVM classifier is devised for emotion recognition. Here Chinese Speaker-dependent and Berlin Speaker-independent speech corpora are used for experimental study. Meanwhile, the mean of the log-spectrum (MLS) is particularly used to improve the feature driven SVM classifier. Since SVM isn't used to recognize multiple emotions directly. Therefore, we set ordinary binary SVM classifier as a contrast experiment to feature driven hierarchical SVM. Then we calculated the recognition rate respectively and analyzed the potential problems. However, the following problems still need further study. (1) All global feature parameters are extracted through the same statistical features, there may be conflict between statistical features and some feature's impact may be reduced when reducing dimensions through PCA. It is a research direction to use other methods such as SFS algorithm to reduce the dimensions. (2) How to extract effective feature parameters. The feature parameters extracted in this paper still can't separate anger, happiness from amazement and sadness from neutral very clearly. (3) Linear kernel function is utilized for binary SVM in this paper, for better performance, new kernel paper may be tried.

## Acknowledgements

This paper was supported by the National Natural Science Foundation of Hunan (13JJ4018, 13JJ4093), the Fundamental Research Funds for the Central Universities (2012QNZT060), and the youth Foundation of education bureau of Hunan province (11B070).

## References

- [1] N. Kamaruddin, A. Wahab, Ch. Quek (2012). Cultural dependency analysis for understanding speech emotion, *Expert Systems with Applications*, 39: 5115-5133.
- [2] M. E. Ayadi, M. S. Kamel, F. Karray (2011). Survey on speech emotion recognition Features, classification schemes and databases, *Pattern Recognition*, 44: 572-587.
- [3] W. Han, H. F. Li (2009). Speech emotion recognition based on prosodic segment level features, *Journal of Tsinghua University (Science and Technology)*, S1:1363-1368.
- [4] T. Iliou, C.N. Anagnostopoulos (2010). SVM-MLP-PNN classifiers on speech emotion recognition field-A comparative study, *The Fifth International Conference on Digital Telecommunications*, Athens/Glyfada, Greece.
- [5] B. Schuller, G. Rigoll, M. Lang (2004). Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture, *IEEE International Conference on Acoustics, Speech, and Signal Processing*.
- [6] E. Bozkurt, E. Erzin, C. E. Erdem, A. T. Erdem (2011). Formant position based weighted spectral features for emotion recognition, *Speech Communication*, 53:1186-1197.
- [7] B. Yang, M. Lugger (2010). Emotion recognition from speech signals using new harmony features, *Signal Processing*, 90:1415-1423.
- [8] M. Sheikhan, M. Bejani, D. Gharavian (2012). Modular neural-SVM scheme for speech emotion recognition using ANOVA feature selection method, *Neural Comput & Applic*, 8:1-13.
- [9] S. Chandaka, A. Chatterjee, S. Munshi (2009). Support vector machines employing cross-correlation for emotional speech recognition, *Measurement*, 42:611-618.
- [10] Burkhardt, F. Paeschke, A. Rolfes, M. Sendlmeier, W. Weiss, B. (2005). A database of German emotional speech, *The 9th European Conference on Speech Communication and Technology*, Euro speech Inter speech Lisbon, Portugal.
- [11] J. Platt, N. Cristianini, J. Shawe-Taylor (2000). Large margin DAGs for multi-class classification, *Proceedings of Neural Information Processing Systems*, Denver.
- [12] A. Hassan, R. I. Damper (2012). Classification of emotional speech using 3DEC hierarchical classifier, *Speech Communication*, 24:1-10.
- [13] K. C. Huang, Y. H. Kuo (2010). A Novel Objective Function to Optimize Neural Networks for Emotion Recognition from Speech Patterns, *The Second World Congress on Nature and Biologically Inspired Computing*, Kitakyushu, Fukuoka, Japan.
- [14] E. M. Albornoz, D. H. Milone, H. L. Rufiner (2011). Spoken emotion recognition using hierarchical classifiers, *Computer Speech and Language*, 25:556-570.
- [15] H.B. Yao, L. Tian (2003). A genetic-algorithm-based selective principal component analysis (GA-SPCA) method for high-dimensional data feature extraction, *IEEE Transactions on Geoscience and Remote Sensing*, 41: 1469-1478.

Lingli YU is working in the institute of Intelligence Science and Technology (IST), School of Information Science and Engineering of Central South University, Changsha, 410083, P.R. China. Her current research interests include speech emotion recognition, mobile-robot fault diagnosis. E-mail: llyu@csu.edu.cn.

Binglu WU is studying in the institute of Intelligence Science and Technology (IST), School of Information Science and Engineering of Central South University, Changsha, 410083, P.R. China. Her current research interests include speech emotion recognition. E-mail: wubingluqipu@163.com.

Tao Gong is Associate Professor of Immune Computation at the College of Information Science and Technology, Donghua University. He holds Top Award of Baosteel Education Fund for his work on immune computation. Tao is the founder and General Editor-in-Chief (EiC) of Immune Computation, which is the first leading journal in the field of immune computation. Tao also acts as a Vice-Chair of IEEE Computer Society Task Force on Artificial Immune Systems. Tao is a Life Member of Sigma Xi, The Scientific Research Society and Shanghai Chenguang Scholar. Tao has worked as the chairs and committee members of many international conferences on artificial immune systems and computer systems.

# Comparing Reinforcement Learning and Evolutionary Based Adaptation in Population Games

Ana L. C. Bazzan

PPGC / UFRGS

Caixa Postal 15064, CEP 91501-970, Porto Alegre, RS, Brazil  
bazzan@inf.ufrgs.br

## Abstract

In evolutionary game theory, the main interest is normally on the investigation of how the distribution of strategies changes along time and whether an stable strategy arises. In this paper we compare the dynamics of two games in which three populations of agents interact: a three-player version of matching pennies and a game with several Nash equilibria. We do this comparison by three methods: continuous replicator dynamics, an evolutionary approach, and reinforcement learning. We show how the convergence depends on the nature of the underlying method used, as well as on the pace of adjustments by the agents.

## Introduction

In game theory (GT), one traditional explanation of equilibrium is that it results from an introspective analysis by the players when the rules of the game, the rationality of players, and the payoff functions are all common knowledge. It is well-known that both conceptually and empirically this argument has many problems. Just to mention one of them, in games with more than one equilibria, even if one assumes that players are able to coordinate their expectations using some selection procedure, it is not clear how such a procedure comes to be common knowledge (Fudenberg and Levine, 1998). Moreover, in the real world, individuals, as for instance animals, are not necessarily rational as assumed by the GT. Thus, in the context of evolutionary GT (EGT), alternative explanations focus on equilibrium arising as a long-run outcome of a process in which populations of animals interact over time.

In the next section, we briefly review some of these approaches. In particular, we note that one of the approaches, the replicator dynamics (RD), presents some problems related to its justification among populations of animals. In fact, there has been a great deal of questioning about why should one care about using the RD, since neither economic agents nor artificial agents (and actually not even monkeys) are genetically programmed to "play" certain behaviors. Thus a justification for the replicator could be that there is an underlying model of learning (by the agents) that gives rise to the dynamics.

Some alternatives have been proposed in this line. Fudenberg and Levine (1998) refer to two interpretation that relate to learning. The first is social learning, a kind of "asking around" model in which players can learn from others in the population. The second is a kind of satisficing-rather-than-optimizing learning process in which the probability of a determined strategy is proportional to the payoff difference with the mean expected payoff.

This second variant has been explored, among others, by Börgers and Sarin (1997); Tuyls et al. (2006), which are based on some kind of reinforcement learning at individual level. In particular, in the stimulus-response based approach of Börgers and Sarin, the reinforcement is proportional to the realized payoff (in their formulation, necessarily positive). Thus each strategy's probability increases by a factor that is computed by the current probability multiplied by the difference between the strategy's expected payoff and the expected payoff of the player's current mixed strategy. In the limit, it is shown that the trajectories of the stochastic process converges to the continuous RD. This is valid in a stationary environment. However, as noted by Börgers and Sarin and Fudenberg and Levine, this does not imply that the RD and the stochastic process have the same asymptotic behavior when the play of *both* players follow a stimulus-response learning approach. We remark that Fudenberg and Levine (1998) specifically mention a two agent or two population game, but the same is true (even more serious), when it comes to more.

The reasons for this difference are manifold. First, Börgers and Sarin's main assumption is "...an appropriately constructed continuous time limit", i.e., a gradual (very small) adjustment is made by players between two iterations of the game. This implies that the discrete learning model evolves stochastically, whereas the equations of the RD are deterministic. Also, there is the fact that players may be stuck in suboptimal strategies because they are all using a learning mechanism, thus turning the problem non-stationary. These facts have as consequences that other popular dynamics in game-theory as, e.g., best response dynamics, which involve instantaneous adjustments to best replies,

have difference in the asymptotic behavior. For example, in the matching pennies game, “discrete time replicator dynamics will cycle along expanding trajectories, but will not get absorbed by any pure strategy outcome” (Börgers and Sarin, 1997). In the battle of the sexes game, a learning approach which is not based on the gradual adaptation (such as best response) may prevent convergence to an equilibrium while the gradual adjustment of RD permits such convergence.

In summary, there are advantages and disadvantages in using the discussed approaches and interpretations, i.e., the continuous, analytical variant of RD, and learning and adaptation approaches such as best response, genetic algorithms, and stimulus-response based models. Besides, as we verify, not all adaptation methods do replicate the RD.

In this paper we aim at applying different approaches and compare their performance. Specifically, we use three of them: the analytical equations of the RD, a genetic based evolutionary approach, and reinforcement learning (here Q-learning). We remark that, as discussed in the next section, some learning approaches are not appropriate for this problem as they either consider perfect monitoring (observation of other individuals’ actions, as in Claus and Boutilier (1998)), or modeling of the opponent (as in fictitious play), or both. In our case this is not possible given the high number of individuals involved in the populations, and the unlikelihood of encounters happening frequently among the same individuals.

Further, we employ two games that are played by three populations of individuals and have been seen as metaphors for studying interactions in populations of individuals. The first is a three-person version of the matching pennies, due to Jordan (apud Fudenberg and Tirole (1991)). In this, as in the two-person original game, there is just a single Nash equilibrium (in mixed strategies). The second is due to B. O’Neill (apud Myerson (2002)), which pays a non-zero quantity to the players when exactly one of them select one of the two strategies. This game has eight Nash equilibria, four of which are in pure strategies, thus making the task of coordinating which equilibrium to select very hard for the individuals.

We are interested in the trajectory of a population of individuals with very little knowledge of the game. Indeed, they are only aware of the payoff or reward received, thus departing from the assumption of knowledge of payoff matrix and rationality being common knowledge, frequently made in GT. We show that, in the case of the three-player matching pennies, the evolutionary approach leads to the same result as the RD, namely to cycle, which means to the (unique) Nash equilibrium in mixed strategy. For the second game, it was possible to observe convergence to the Nash equilibria in pure strategies. In all cases, convergence depends on the rate of experimentation in the populations.

## Background and Related Work

### Evolutionary Game Theory

EGT investigates the relationship between individual and aggregate behaviors. Its inspiration comes from population genetics, where the focus is less on individual behavior and more on the aggregate population behavior. This shift from individual-level decision-making (eventually leading to user equilibrium), to dynamics of individuals interaction is in line with the increasing complexity in modern societies. There are many systems where we nowadays observe a tendency of a complex coupled decision-making process. Already in 1950, Nash saw this phenomenon, which he then called “mass-action interpretation”. Later, this focus on equilibria was criticized by J. Maynard Smith: “An obvious weakness of the game-theoretic approach to evolution is that it places great emphasis on equilibrium states ...” (Smith, 1982). Besides, J. Maynard Smith also dealt with the shift from individual to population level. Even if he borrowed some definitions from standard GT when he introduced the concept of evolutionary stable strategy (ESS) as a way to understand conflicts among animals, he had already noticed that “there are many situations ... in which an individual is, in effect, competing not against an individual opponent but against the population as a whole... Such cases can be described as ‘playing the field’ ...” (Smith, 1982).

Currently, this kind of modeling is called a population game, which models simultaneous interactions of a large number of simple individuals or agents distributed in a finite number of populations. Simple agents here mean that each has a (typically small) number of strategies to choose, causing a minor impact in other agents payoff. Despite this, the payoff of each agent is, as in the classical GT, conditioned by the distribution of strategies in each population. In EGT and population games, typically, one is not interested in constancy or equilibrium only. Rather, the major interest is on the dynamics of games. A population of decision-makers is considered, in which what is investigated is how the rate of the strategy profiles change as a response to the decisions made by all individuals in the population.

This idea that the composition of the population of individuals (and hence of strategies) in the next generations changes with time (in this case generations) suggests that we can see these individuals as replicators. The RD is based on gradual movement from worse to better strategies. One of the results of Börgers and Sarin is that in appropriately constructed continuous time limit, a stimulus-response based learning model converges to the continuous time version of the RD. They have proposed that such a continuous time limit is constructed so that each time interval sees many iterations of the game, and that the adjustments that the agents make between two iterations of the game are very small. This way the stochastic learning process becomes deterministic in the limit, thus replicating the system of differential equations which characterizes the RD.



However, as mentioned in the introduction, this result refers to arbitrary, finite points in time, and does not hold if infinite time is considered. When time tends to infinitely, the asymptotic behavior of the discrete time learning process can be different from the asymptotic behavior of the continuous time RD.

Additionally, the RD treats the player as a population (of strategies). By the construct of the continuous time of Börgers and Sarin, in each iteration, a random sample of the population is taken to play the game. Due to the law of large numbers, this sample represents the whole population. However, in the discrete learning process, at each time, only one strategy is played by each individual. Moreover, the outcome of each of these interactions affects the probabilities with which the strategies are used in the next time step.

These results have been extended in Tuyls et al. (2006). It was shown that the positive reinforcement model by Börgers and Sarin (1997) corresponds to the learning automata. Moreover, a similar dynamics was derived for Boltzmann action selection. The theoretical results were verified with experiments in 3 classes of  $2 \times 2$  games.

These works suggest that other dynamics, e.g., based on less gradual adjustments may lead to different results in other games as well. We also remark that at least one of the games considered in this paper is one in which not only the analytical computation of the RD is non-trivial, but also the fact that more populations and actions are involved contribute to results being less intuitive as the cases in Börgers and Sarin (1997); Tuyls et al. (2006).

Next, for sake of clarity, we briefly mention the Q-learning method, as well as discuss some related work on multiagent reinforcement learning (MARL).

### Individual and Multiagent RL

Reinforcement learning (RL) by a single agent problems can be modeled as Markov decision processes (MDPs). An experience tuple  $\langle s, a, s', r \rangle$  denotes the fact that the agent was in state  $s$ , performed action  $a$  and ended up in  $s'$  with reward  $r$ . Q-learning is a popular model-free algorithm in which the update rule for each experience tuple is given in Equation 1 where  $\alpha$  is the learning rate and  $\gamma$  is the discount for future rewards.

$$Q(s, a) \leftarrow Q(s, a) + \alpha (r + \gamma \max_{a'} Q(s', a') - Q(s, a)) \quad (1)$$

When many individuals or agents learn simultaneously, the problems are well known. They arise mainly due to the fact that while one agent is trying to model the environment (other agents included), the others are doing the same and potentially changing the environment they share.

Besides, in MARL an issue is the exponential increase in the space of joint states and joint actions, if agent  $i$  *explicitly models the states and actions of other agents*. The decision

on whether or not to include joint states and/or joint actions in the learning process of  $i$  is a key one as it has severe implications. In fact, most of the game-theoretic literature concentrates on games with few players and few actions because otherwise it is computationally prohibitive.

This paper deals with a large number of agents, thus approaches such as JAL (joint agent learners, Claus and Boutilier (1998)) cannot be used because an explicit model of other agents' actions, states, and rewards is necessary. After JAL, several approaches have been proposed for related as well as more general MARL problems. However, they cannot be used due to some restrictions: zero-sum game (e.g., Littman (1994)), few agents and/or few actions and/or assumption of perfect monitoring (e.g., Hu and Wellman (1998); Lauer and Riedmiller (2000); Kapetanakis and Kudenko (2002); Kuminov and Tennenholtz (2008)).

## Methods

### Formalization of Population Games

Population games are quite different from the games studied by the classical GT because population-wide interaction generally implies that the payoff to a given member of the population is not necessarily linear in the probabilities with which pure strategies are played. A population game can be defined as follows.

- **(populations)**  $\mathcal{P} = \{1, \dots, p\}$ : society of  $p \geq 1$  populations of individuals where  $|p|$  is the number of populations
- **(strategies)**  $\mathcal{S}^p = \{s_1^p, \dots, s_m^p\}$ : set of strategies available to agents in population  $p$
- **(payoff function)**  $\pi(s_i^p, \mathbf{q}^{-p})$

Agents in each population  $p$  have  $m^p$  possible strategies. Let  $n_i^p$  be the number of individuals using strategy  $s_i^p$ . Then, the fraction of individuals using  $s_i^p$  is  $x_i^p = \frac{n_i^p}{N^p}$ , where  $N^p$  is the size of  $p$ .  $\mathbf{q}^p$  is the  $m^p$ -dimensional vector of the  $x_i^p$ , for  $i = 1, 2, \dots, m^p$ . As usual,  $\mathbf{q}^{-p}$  represents the set of  $\mathbf{q}^p$ 's when excluding the population  $p$ . The set of all  $\mathbf{q}^p$ 's is  $\mathbf{q}$ . Hence, the payoff of an agent of population  $p$  using strategy  $s_i^p$  while the rest of the populations play the profile  $\mathbf{q}^{-p}$  is  $\pi(s_i^p, \mathbf{q}^{-p})$ . Consider a (large) population of agents that can use a set of pure strategies  $\mathcal{S}^p$ . A population profile is a vector  $\sigma$  that gives the probability  $\sigma(s_i^p)$  with which strategy  $s_i^p \in \mathcal{S}^p$  is played in  $p$ .

One important class within population games is the class of symmetric games, in which two random members of a *single* population meet and play the stage game, whose payoff matrix is symmetric. The reasoning behind these games is that members of a population cannot be distinguished, i.e., two meet randomly and each plays one role but these need not to be the same in each context. Thus the symmetry. However, there is no reason to restrict oneself to a symmetric modeling in other scenarios beyond population biology. For

|   |  | H        |          |
|---|--|----------|----------|
|   |  | H        | T        |
| H |  | +1/+1/-1 | -1/-1/-1 |
| T |  | -1/+1/+1 | +1/-1/+1 |

|   |  | T        |          |
|---|--|----------|----------|
|   |  | H        | T        |
| H |  | +1/-1/+1 | -1/+1/+1 |
| T |  | -1/-1/-1 | +1/+1/-1 |

Table 1: Payoff matrices for the three-player matching pennies game; payoffs are for player 1 / player 2 / player 3.

|       |  | $x_3$        |              |
|-------|--|--------------|--------------|
|       |  | $x_2$        | $y_2$        |
| $x_1$ |  | 0/0/0        | <b>6/5/4</b> |
| $y_1$ |  | <b>5/4/6</b> | 0/0/0        |

|       |  | $y_3$        |              |
|-------|--|--------------|--------------|
|       |  | $x_2$        | $y_2$        |
| $x_1$ |  | <b>4/6/5</b> | 0/0/0        |
| $y_1$ |  | 0/0/0        | <b>0/0/0</b> |

Table 2: Payoff matrices for the 3PEOY game; payoffs are for player 1 / player 2 / player 3 (the four Nash equilibria in pure strategies are indicated in boldface).

instance, in economics, a market can be composed of buyers and sellers and these may have asymmetric payoff functions and/or may have sets of actions whose cardinality is not the same. In asymmetric games, each agent belongs to one class determining the set of legal strategies.

Before we present the particular modeling of asymmetric population game, we introduce the concept of RD.

The previously mentioned idea that the composition of the population of individuals (and hence of strategies) in the next generations changes with time suggests that we can see these individuals as replicators. In the RD, the rate of use of a determined strategy is proportional to the payoff difference with the mean expected payoff, as in Eq. 2. As previously defined, the fraction of agents using  $s_i^p$  is  $x_i^p = \frac{n_i^p}{N^p}$ . The state of population  $p$  can be described as a vector  $\mathbf{x}^p = (x_1^p, \dots, x_m^p)$ . We are interested in how the fraction of agents using each strategy changes with time, i.e., the derivative  $\frac{dx_i^p}{dt}$  (henceforth denoted  $\dot{x}_i^p$ ).

$$\dot{x}_i^p = (\pi(s_i^p, \mathbf{x}^p) - \bar{\pi}(\mathbf{x}^p)) \times x_i^p \quad (2)$$

In Eq. 2,  $\bar{\pi}(\mathbf{x}^p)$  is the average payoff obtained by  $p$ :

$$\bar{\pi}(\mathbf{x}^p) = \sum_{i=1}^m x_i^p \pi(s_i^p, \mathbf{x}^p)$$

Obviously, to analytically compute this average payoff, each individuals would have to know all the payoffs, which is quite unrealistic.

## Two Scenarios for Three Player Games

In the three-population games considered here, to avoid confusion we use the term "player" with its classical interpretation, i.e., the decision-makers of the normal form game (NFG). Because this game is played by randomly matched individuals, one from each population, we call these individuals "agents". Thus player refers to a population of agents.

The just given description of the general population game is instantiated for our particular scenarios as follows.

**Three Player Matching Pennies** From the general definition of a population game given in the previous section, this is the particular instance for the three player matching pennies (henceforth 3PMP):

- **(populations)**  $\mathcal{P} = \{1, 2, 3\}$
- **(strategies)** for each population  $p \in \mathcal{P}$ :  $\mathcal{S}^1 = \mathcal{S}^2 = \mathcal{S}^3 = \{H, T\}$
- **(payoff function)** see Table 1

From the payoff matrix for this game (Table 1), one sees that as in the two player original game, there is a single Nash equilibrium, in mixed strategies. This is indicated in Table 3. In this table, columns 2–3 specify  $\mathbf{x}^1$  (fraction of agents selecting each strategy H and T in population  $p = 1$ ), columns 4–5 specify  $\mathbf{x}^2$  of  $p = 2$ , and columns 6–7 specify  $\mathbf{x}^3$  of  $p = 3$ . The last column gives the payoffs for the agents.

**Three Player Coordination Game** The second three player game that we consider is a kind of coordination game. As mentioned, it pays a non-zero quantity to the players when exactly one of them select one of the two strategies. Henceforth we denominate this game by 3PEOY (three players, where exactly one should select strategy  $y$ ). This game has eight Nash equilibria, four of which are in pure strategies, thus making the task of coordinating over equilibrium selection very hard for the agents.

The corresponding population game is then defined:

- **(populations)**  $\mathcal{P} = \{1, 2, 3\}$
- **(strategies)** for each population  $p \in \mathcal{P}$ :  $\mathcal{S}^1 = \{x_1, y_1\}$ ,  $\mathcal{S}^2 = \{x_2, y_2\}$ , and  $\mathcal{S}^3 = \{x_3, y_3\}$ .
- **(payoff function)** see Table 2

| profile  | $x_H^1$       | $x_T^1 = (1 - x_H^1)$ | $x_H^2$       | $x_T^2 = (1 - x_H^2)$ | $x_H^3$       | $x_T^3 = (1 - x_H^3)$ | payoff |
|----------|---------------|-----------------------|---------------|-----------------------|---------------|-----------------------|--------|
| $\sigma$ | $\frac{1}{2}$ | $\frac{1}{2}$         | $\frac{1}{2}$ | $\frac{1}{2}$         | $\frac{1}{2}$ | $\frac{1}{2}$         | 0/0/0  |

Table 3: Unique Nash equilibria for the 3PMP game.

| profile    | $x_x^1$        | $x_y^1 = (1 - x_x^1)$ | $x_x^2$        | $x_y^2 = (1 - x_x^2)$ | $x_x^3$        | $x_y^3 = (1 - x_x^3)$ | payoff                                       |
|------------|----------------|-----------------------|----------------|-----------------------|----------------|-----------------------|----------------------------------------------|
| $\sigma_a$ | 1              | 0                     | 1              | 0                     | 0              | 1                     | 4/6/5                                        |
| $\sigma_b$ | 0              | 1                     | 0              | 1                     | 0              | 1                     | 0/0/0                                        |
| $\sigma_c$ | 0              | 1                     | 1              | 0                     | 1              | 0                     | 5/4/6                                        |
| $\sigma_d$ | 1              | 0                     | 0              | 1                     | 1              | 0                     | 6/5/4                                        |
| $\sigma_e$ | 1              | 0                     | $\approx 0.44$ | $\approx 0.56$        | $\approx 0.54$ | $\approx 0.45$        | $\approx 2.63 / \approx 2.73 / \approx 2.22$ |
| $\sigma_f$ | $\approx 0.44$ | $\approx 0.56$        | $\approx 0.54$ | $\approx 0.45$        | 1              | 0                     | $\approx 2.73 / \approx 2.22 / \approx 2.63$ |
| $\sigma_g$ | $\approx 0.54$ | $\approx 0.45$        | 1              | 0                     | $\approx 0.44$ | $\approx 0.56$        | $\approx 2.22 / \approx 2.63 / \approx 2.73$ |
| $\sigma_h$ | $\frac{2}{3}$  | $\frac{1}{3}$         | $\frac{2}{3}$  | $\frac{1}{3}$         | $\frac{2}{3}$  | $\frac{1}{3}$         | $\approx 2.22 / \approx 2.22 / \approx 2.22$ |

Table 4: Eight Nash equilibria for the 3PEOY game.

The eight Nash equilibria appear in Table 4. In this table, columns 2–3 specify  $\mathbf{x}^1$  (fraction of agents selecting each strategy  $s_i^1$  in population  $p = 1$ ), columns 4–5 specify  $\mathbf{x}^2$  of  $p = 2$ , and columns 6–7 specify  $\mathbf{x}^3$  of  $p = 3$ . The last column gives the payoffs for the agents. For example, for the first equilibrium (profile  $\sigma_a$ ), because  $x_x^1 = 1$ ,  $x_x^2 = 1$ , and  $x_x^3 = 0$ , all agents in  $p = 1$  select action  $x_1$ , all agents in  $p = 2$  select  $x_2$  and all agents in  $p = 3$  select  $y_3$ .

Regarding the mixed strategy profile  $\sigma_e$  (for example), all agents in  $p = 1$  select action  $x_1$  (because  $x_y^1 = 0$ ), whereas in the other two populations nearly half of the agents select each action. Profiles  $\sigma_f$  to  $\sigma_h$  can be similarly interpreted.

In the classical GT interpretation of equilibrium, profiles  $\sigma_a$ ,  $\sigma_b$ ,  $\sigma_c$ , and  $\sigma_d$  would be Nash equilibria in pure strategies, while the other four equilibria would mean that agents randomize between at least two pure strategies. The EGT interpretation thought is as follows. If we consider that we are dealing with three populations of agents, we can think about the equilibria in terms of the percentage of individuals in one of the three populations that in fact select one of the actions available. This seems a more reasonable explanation for the concept of mixed strategies, given that, at each time, agents in fact only select an action.

The same reasoning obviously applies to the 3PMP game which has only a mixed strategy profile.

Moreover, it must also be noticed that in asymmetric games, all ESS are pure strategies (for a proof see, e.g., Webb (2007)). Thus, for the 3PEOY game, only  $\sigma_a$  to  $\sigma_d$  are candidates for ESS.

Note from Table 4 that in this game  $\sigma_b$ , though a Nash equilibrium, is not efficient (all agents receive zero), thus the learning models tested here should be able to recognize this.

### Replicator Dynamics, Evolutionary Approach and Reinforcement Learning

As mentioned in the introduction, the continuous RD model, which is hard to justify, can be reproduced with some forms of learning. To compare the performance of these learning models, we have first formulated the continuous RD for our specific three-population game. The equations can be derived from Eq. 2.

In both the 3PMP and 3PEOY, we have checked which Nash equilibria are stable. In the 3PEOY in particular, the Nash equilibria that need to be investigated are, as mentioned, those in pure strategies, i.e.,  $\sigma_a$  to  $\sigma_d$  from Table 4. To check which are the ESSs, it is necessary to analyze which are the stable rest points. In simple games, e.g.,  $2 \times 2$  or even symmetric  $2 \times 3$  this can be done graphically. However, because our problem involves several variables the divergence operator was used.

Now we turn to the approaches based on evolution and reinforcement learning. In both cases, in each time step, agents from each population  $p$  play  $g$  games whose payoffs are given in Table 1 and Table 2.

For the evolutionary approach, mutations create genetically mutated versions of the agents. Here we use a mutation rate  $p_m$ : at each time step, each agent in the population receives a new strategy with probability  $p_m$ . We recall that, according to Börgers and Sarin (1997), it is expected that if the adjustment is not gradual, there may be no convergence to the behavior of the continuous RD. The sum of the payoffs obtained by playing these  $g$  games is then the fitness of the agent. After these  $g$  games are played, the populations of agents are reproduced: In each population  $p$ , the fittest agents have a higher probability of being selected. Then each individual suffers mutation with probability  $p_m$ , which means that its strategy is changed to another one randomly selected.

For the reinforcement learning, agents learn using individual Q-learning (Eq. 1), thus assessing the value of each strategy by means of Q values. For action selection,  $\epsilon$ -greedy was used, i.e., action selection is random with probability  $\epsilon$ , otherwise it is greedy. In line with the just mentioned issue of gradual adjustments, and from Tuyls et al. (2006), we know that the value of  $\epsilon$  is key to reproduce the behavior of the continuous RD.

## Experiments and Results

In this section we discuss the numerical simulations of the evolutionary and learning based approaches and compare them with the continuous RD, from which we know the Nash equilibria, and the candidates to be ESSs, for both games 3PMP and 3PEOY.

We are interested in investigating issues such as what happens if each population  $p$  starts using a given profile  $\sigma^p$  in games that have more than one equilibrium. To which extent the rate  $p_m$  shifts this pattern? For instance, if the population starts using any  $\sigma^p$ , what happens if it is close to (but not actually at)  $\sigma^*$ ? Will it tend to evolve towards  $\sigma^*$  or move away? If it reaches  $\sigma^*$ , how long has it taken? What happens if there are multiple equilibria?

The main parameters, as well as the values that were used in the simulations are:  $\mathcal{P} = \{0, 1, 2\}$ ,  $N^0 = N^1 = N^2 = 300$ ,  $g = 10,000$ ,  $\alpha = 0.5$ ;  $\epsilon$ ,  $\Delta$  (number of time steps) and  $p_m$  were varied. In all cases, at step 0, agents select strategies from a uniform distribution of probability.

The next subsections discuss the evolutionary and the reinforcement learning approaches respectively.

### Adaptation with Evolutionary Approach

In this case, because more than two variables (strategies) are involved, it is not possible to show typical (2d) RD-like plots that depict the trajectory of these variables. Therefore, as an alternative to show the dynamics, we use heatmaps.

We start with the 3PMP game. In the plots that appear in Figure 1 (which were reduced due to lack of space), heatmaps are used to convey the idea of the intensity of the selection of each of the 8 joint actions (represented in the vertical axis) along time (horizontal axis), with  $\Delta = 1000$  time steps. These 8 joint actions are those that appear in Table 1. Due to an internal coding used, the 8 joint actions are labeled such that 0 and 1 mean the selection of first strategy ( $H$ ) and second strategy ( $T$ ) respectively.

In each triplet (joint action), the first digit indicates the action of  $p = 3$ , the second digit is for the action of  $p = 2$ , and the third digit is for  $p = 1$ .

In the heatmaps, to render the figure cleaner we just use shades of gray color (instead of hot colors as usual). In any case, the darker the shade, the more intense one joint action is selected. Thus we should expect that the Nash equilibria correspond to the darker strips.

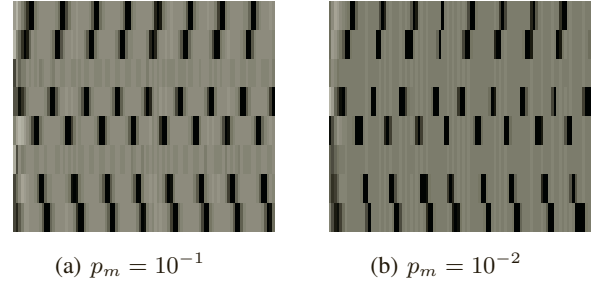


Figure 1: 3PMP: Evolution of Dynamics for Different values of  $p_m$ , evolutionary approach.

In Figure 1 it is possible to see that there is no convergence to any of the pure strategy. Rather, agents cycle among 6 of the joint actions. This happens for all value of  $p_m$  tested. The two joint actions that pay  $-1$  to all agents are quickly discarded. Remember that a different pattern occurs when fictitious play was used, according to Börgers and Sarin (1997).

Regarding the second game, 3PEOY, plots appear in Figure 2. Again, there are 8 joint actions. These are labeled such that 0 and 1 mean the selection of first strategy ( $x$ ) and second strategy ( $y$ ) respectively.

In particular, the four Nash equilibria ( $\sigma_a$ ,  $\sigma_b$ ,  $\sigma_c$ , and  $\sigma_d$ ) in pure strategies of these game are represented as: 100, 111, 001 and 010 respectively.

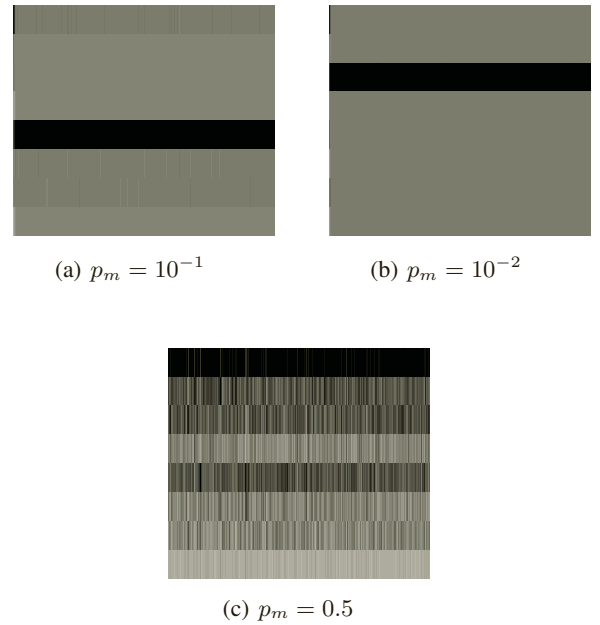


Figure 2: 3PEOY: Evolution of Dynamics for Different values of  $p_m$ , evolutionary approach.



In Figure 2 we show how the selection evolves along time, for two values of  $p_m$ . Figure 2(a) is for  $p_m = 10^{-1}$ , i.e., changes in strategy occur with this rate. In these particular plot we show clear convergence to  $\sigma_a$  (100). However, convergence is also possible to other joint actions, as, e.g. 010 and 001, but not to 111 because this is a non efficient Nash equilibria. We do not show all plots as they are similar to Figure 2(a).

When  $p_m$  is lowered as in Figure 2(b), the pattern is the same, i.e., clear convergence to either  $\sigma_a$ ,  $\sigma_c$  or  $\sigma_d$ .

When  $p_m$  is set higher, this pattern does not occur, i.e., there is no clear convergence to any of the Nash equilibria. An extreme case with  $p_m = 0.5$  is depicted in Figure 2(c), where one sees that the performance is poor due to the high rate of changes by the agents. Interestingly, frequently, the Pareto inefficient Nash equilibria  $\sigma_b$  is selected, which means payoff of zero to each agent.

### Reinforcement Learning

We now turn to the individual learning using Q-learning. Experiments were run with different values of  $\alpha$  and change in  $\varepsilon$ . It seems that  $\alpha$  has much less influence in the result than  $\varepsilon$ . Thus we concentrate on  $\alpha = 0.5$ .

For both games, here we show plots for  $\varepsilon$  starting at 1.0 with various decay rates each time step.

For the 3PMP game, to render the picture more clear, plots in Figure 3 depict the evolution within time of the percentage of H being selected by agents in population  $p = 1$  only as others are similar. One can see that agents shift from H to T and vice-versa. This happens no matter if the decay of  $\varepsilon$  is fast (Figure 3(a)) or more slow (Figure 3(b)).

For the 3PEOY game, due to a more clear convergence pattern, it is possible to plot not only the percentage of selection of action  $x$  by agents in the first population, but also for  $p = 2$  and  $p = 3$ . Figure 4 depicts them. Figure 4(a) refers to a faster decay of  $\varepsilon$ , 0.9 at each time step, and shows convergence to  $\sigma_c$ . However we note that similar patterns occur leading to  $\sigma_a$  and  $\sigma_d$  (but not  $\sigma_b$ ). Figure 4(b) is for decay of 0.99. In this particular plot, convergence is to  $\sigma_d$ .

How agents have converged to a given profile is better seen examining the trajectories of the probabilities to play each strategy, for each population. Due to the number of variables, it is not possible to plot them all together. Thus we opted to show the behavior of selected variables in a pairwise fashion, namely  $x_1 \times x_2$ ,  $x_1 \times x_3$ ,  $x_2 \times x_3$  (Figure 5). In this plot, for each of the three pairs, the  $x$ -axis represents the probability with which the first component of the pair selects strategy  $x$ , while the  $y$ -axis represents the probability with which the second component of the pair selects  $x$ .

It is possible to see that although these percentages all start at 0.5, they all converge to 1.0 or close. In the particular plot convergence was to  $x_1 = 1$ ,  $x_2 = 0$ , and  $x_3 = 1$  and hence this corresponds to the same pattern as in Figure 4(b),  $\sigma_d$ .

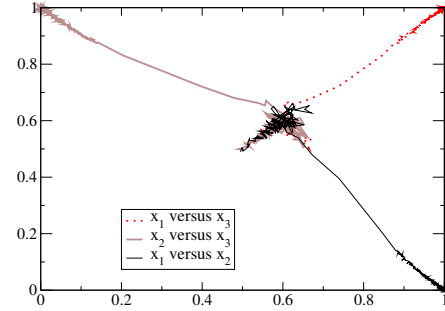


Figure 5: Trajectories:  $x_1 \times x_2$ ,  $x_1 \times x_3$ , and  $x_2 \times x_3$  (all start at  $x_i = 0.5$ ).

In short, some conclusions can be drawn from these simulations. First, simultaneous learning by the agents does not always lead to the Nash equilibrium, much less to the ESS computed for the corresponding RD of the static NFG.

If an evolutionary approach is used, depending on the  $p_m$  rate, any of the Nash equilibria may establish, or agents may be stuck at a sub-optimal state.

This is in line with the result in Börgers and Sarin (1997), which prescribes gradual adjustments. Profiles that are dominated do not establish.

### Conclusion

In this paper, two three-population games were used to illustrate the patterns of convergence when different methods are used for replicating the dynamics prescribed by analytical, continuous methods such as the RD. It was seen that the extent of match between the continuous RD and discrete learning methods (whether an evolutionary approach or Q-learning) depends on the pace of the adjustment.

Also, compared to results in Tuyls et al. (2006), an extra population in the matching pennies game has slowed down the convergence as it takes more time for agents to learn in the presence of three populations.

Future work will consider information broadcast to agents, in order to have a further way to model action selection and try to improve the coordination task, as well as investigate issues regarding correlated equilibria.

### Acknowledgements

The author and this work are partially supported by CNPq.

### References

- Börgers, T. and Sarin, R. (1997). Learning through reinforcement and replicator dynamics. *Journal of Economic Theory*, 77(1):1–14.
- Claus, C. and Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of*

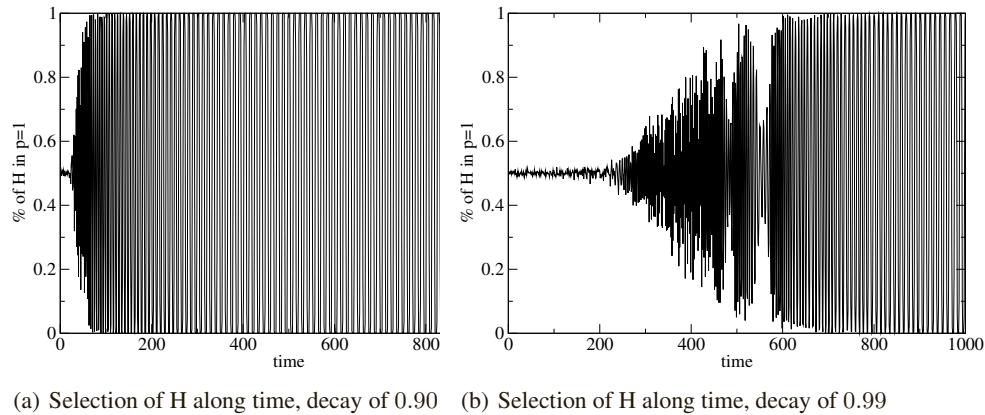


Figure 3: Evolution of the selection of the first player in the 3PMP game, QL.

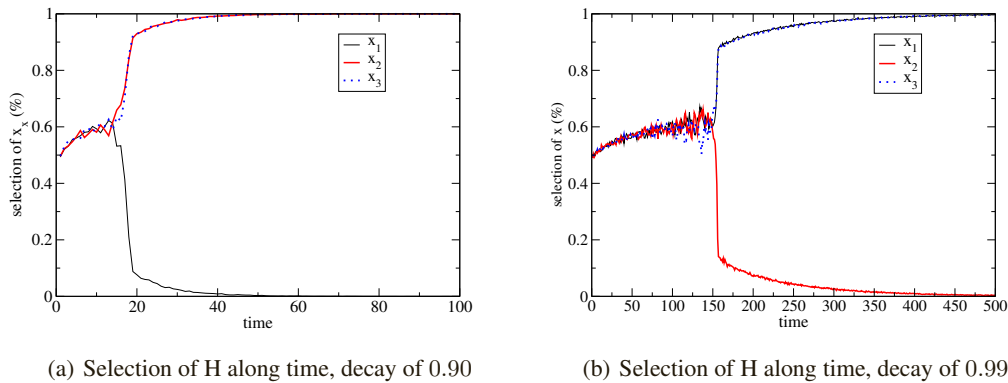


Figure 4: Evolution of the selection of the three players in the 3PEOY game, QL.

the Fifteenth National Conference on Artificial Intelligence, pages 746–752.

Fudenberg, D. and Levine, D. K. (1998). *The Theory of Learning in Games*. The MIT Press.

Fudenberg, D. and Tirole, J. (1991). *Game Theory*. MIT Press Books. The MIT Press.

Hu, J. and Wellman, M. P. (1998). Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proc. 15th International Conf. on Machine Learning*, pages 242–250. Morgan Kaufmann.

Kapetanakis, S. and Kudenko, D. (2002). Reinforcement learning of coordination in cooperative multi-agent systems. In *AAAI/IAAI*, pages 326–331.

Kuminov, D. and Tennenholtz, M. (2008). As safe as it gets: Near-optimal learning in multi-stage games with imperfect monitoring. In *Proceeding of the ECAI 2008*, pages 438–442, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Lauer, M. and Riedmiller, M. (2000). An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In

*Proc. 17th International Conference on Machine Learning*, pages 535–542. Morgan Kaufmann, San Francisco, CA.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning, ML*, pages 157–163, New Brunswick, NJ. Morgan Kaufmann.

Myerson, R. (2002). *Game theory*. Harvard Univ. Press, Cambridge, MA, 5. print edition.

Nash, J. (1950). *Non-Cooperative Games*. PhD thesis, Princeton University.

Smith, J. M. (1982). *Evolution and the Theory of Games*. Cambridge University Press, Cambridge, UK.

Tuyls, K., Hoen, P. J., and Vanschoenwinkel, B. (2006). An evolutionary dynamical analysis of multi-agent learning in iterated games. *Autonomous Agents and Multiagent Systems*, 12(1):115–153.

Webb, J. N. (2007). *Game Theory – Decisions, Interaction and Evolution*. Springer, London.

# Improving Grammatical Evolution in Santa Fe Trail using Novelty Search

Paulo Urbano<sup>1</sup> and Loukas Georgiou<sup>2</sup>

<sup>1</sup>LabMAg - FCUL, Lisboa Portugal  
pub@di.fc.ul.pt

<sup>2</sup>luc\_georgiou@hotmail.com

## Abstract

Grammatical Evolution is an evolutionary algorithm that can evolve complete programs using a Backus Naur form grammar as a plug-in component to describe the output language. An important issue of Grammatical Evolution, and evolutionary computation in general, is the difficulty in dealing with deceptive problems and avoid premature convergence to local optima. Novelty search is a recent technique, which does not use the standard fitness function of evolutionary algorithms but follows the gradient of behavioral diversity. It has been successfully used for solving deceptive problems mainly in neuro-evolutionary robotics where it was originated. This work presents the first application of Novelty Search in Grammatical Evolution (as the search component of the later) and benchmarks this novel approach in a well-known deceptive problem, the Santa Fe Trail. For the experiments, two grammars are used: one that defines a search space semantically equivalent to the original Santa Fe Trail problem as defined by Koza and a second one which were widely used in the Grammatical Evolution literature, but which defines a biased search space. The application of novelty search requires to characterize behavior, using behavior descriptors and compare descriptions using behavior similarity metrics. The conducted experiments compare the performance of standard Grammatical Evolution and its Novelty Search variation using four intuitive behavior descriptors. The experimental results demonstrate that Grammatical Evolution with Novelty Search outperforms the traditional fitness based Grammatical Evolution algorithm in the Santa Fe Trail problem demonstrating a higher success rates and better solutions in terms of the required steps.

## Background

### Grammatical Evolution

Grammatical Evolution (O'Neill and Ryan, 2001) is an evolutionary algorithm that can evolve complete programs in an arbitrary language using populations of variable-length binary strings. Namely, a chosen evolutionary algorithm (typically a variable-length genetic algorithm) creates and evolves a population of individuals and the binary string (genome) of each individual determines which production rules in a Backus Naur Form (BNF) grammar definition are used in a genotype-to-phenotype mapping process to generate a program. In natural biology, there is no direct mapping

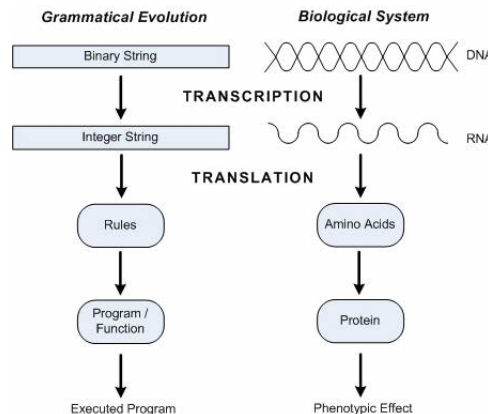


Figure 1: Comparison between the GE system and a biological genetic system. Cited in (O'Neill and Ryan, 2001), p.351.

between the genetic code and its physical expression. Instead, genes guide the creation of proteins, which affect the physical traits either independently or in conjunction with other proteins (Ryan et al., 1998). Grammatical Evolution treats each genotype to phenotype transition as a "protein" which cannot generate a physical trait on its own. Instead, each one protein can result in a different physical trait depending on its position in the genotype and consequently, the previous proteins that have been generated.

Fig. 1 shows the comparison between the GE system and a biological genetic system (O'Neill and Ryan, 2001). The binary string of the genotype of an individual in GE is equivalent to the double helix of DNA of a living organism, each guiding the formation of the phenotype. In the case of GE, this occurs via the application of production rules to generate the terminals of the resulted program (phenotype) and in the biological case, directing the formation of the phenotypic protein by determining the order and type of protein subcomponents (amino acids) that are joined together.

Before the evaluation of each individual, the following steps take place in Grammatical Evolution:

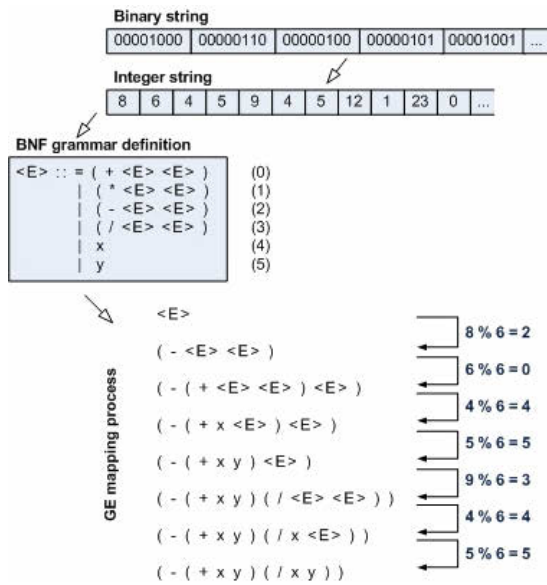


Figure 2: GE mapping process. From Dempsey et al. (2006), p. 2588, (with minor changes)

- i The genotype (a variable-length binary string) is used to map the start symbol of the BNF grammar definition into terminals. The grammar is used to specify the legal phenotypes.
- ii The GE algorithm reads "codons" of typically 8 bits (integer codons are also widely adopted) and the integer corresponding to the codon bit sequence is used to determine which form of a rule is chosen each time a non-terminal to be translated has alternative forms. If while reading "codons", the algorithm reaches the end of the genotype, it starts reading again from the beginning of the genotype (wrapping).
- iii The form of the production rule is calculated using the formula  $form = codon \bmod forms$  where *codon* is the codon integer value, and *forms* is the number of alternative forms for the current non-terminal.

An example of the mapping process employed by Grammatical Evolution is shown in Fig. 2. In this example, the first codon of the genotype of the individual is the binary string 00001000 which is the binary form of the integer 8. The start symbol  $\langle E \rangle$  has six alternative forms, therefore the form to be applied is this with label 2 ( $8 \% 6$ ) which results in the expression  $( - \langle E \rangle \langle E \rangle )$ . The next codon is then read in order to replace the first non-terminal symbol  $\langle E \rangle$  of the new expression, and this goes on until the expression contains only the terminal symbols  $x$  and  $y$  and any of the arithmetic operators. Namely, until all of the non-terminal symbols have been replaced.

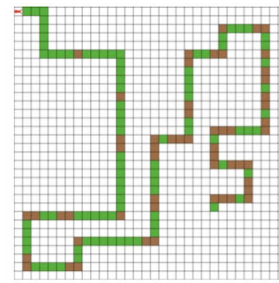


Figure 3: "The Santa Fe Trail.

After the mapping process (i.e. the creation of the phenotype), the fitness score is calculated and assigned to each individual (phenotype) according to the given problem specification. These fitness scores are sent back to the evolutionary algorithm which uses them to evolve a new population of individuals. Grammatical Evolution is a flexible and promising grammar-based evolutionary algorithm with two unique features inspired from molecular biology: genetic code degeneracy which improves genetic diversity, due to its many-to-one genotype-to-phenotype mapping mechanism; and genetic material reuse, due to the genome wrapping it applies. Even though Grammatical Evolution shows competence on a series of problems where it has been applied, the experiments conducted by Georgiou and Teahan (2010, 2011) and Georgiou (2012) cast doubt about its effectiveness and efficiency on deceptive problems such as Artificial Ant and Maze searching.

### The Santa Fe Trail Problem

The Santa Fe Trail is the most common instance of the Artificial Ant problem and was designed by Christopher Langton (Koza, 1991). It is a standard and challenging problem, which is widely used for benchmarking in Genetic Programming (Koza, 1991, 1992) and Grammatical Evolution (O'Neill and Ryan, 2001, 2003).

This problem can be briefly described as finding a computer program to control an artificial ant, within a limited amount of steps, such that it can find all food items forming a twisting trail in a 32 x 32 toroidal plane grid. The trail is composed by 89 food pellets distributed non-uniformly along it, and has 55 gaps and 21 turns (see Fig. 3). The Santa Fe Trail has the following irregularities: single gaps, double gaps, single gaps at corners, double gaps at corners (short knight moves) and triple gaps at corners (long knight moves) (Koza, 1991).

The artificial ant starts in the upper left cell of the grid (0,0), facing east, and can perform three primitive actions: *move*, *turn right* and *turn left* - each action consumes one step. The *move* action moves the ant forward one square in the direction it is currently facing. When the ant moves into a square, it eats the food if there is any. The other two actions



turn the ant right or left respectively by 90 degrees, without moving the ant. The ant can use a binary sensing boolean operator *food ahead*, which comes with no cost in moves. This sensing operator looks into the square the ant is currently facing and executes one of its arguments depending upon whether the square ahead contains food or is empty. Besides the sensing operator, Koza (1991) introduced the unlimited sequence operator *progn* that executes its arguments in order. Alternatively, Koza (1992) introduced two and limited sequence *progn2* and *progn3*, taking two and three arguments respectively. This difference regarding the sequence operators used introduces a representational bias, but does not affect the ant control possibilities, since any *progn* subtree can be translated into semantically equivalent subtrees using *progn2* and *progn3* operators and vice versa (Robilliard et al., 2005).

During the evaluation of the ant, its program is iterated until exhausting the steps limit or eating all food items. There is no unanimity in the literature regarding the fixation of the maximal number of steps an ant is allowed to perform. Robilliard et al. (2005) note that Koza said he arbitrarily fixed to 400, but Langdon and Poli (1998) set the limit to 600 steps, assuming a possible mistype in Koza's paper. O'Neill and Ryan (2003) set the maximum number of steps to 615 and Georgiou (2012) benchmarks a variety of GE configurations using 650 steps.

As the goal is to collect the maximal number of food items laid down in the trail, the customary fitness function is the amount of food collected by the ant, or in the case of minimization, the number of pellets missed out of the total 89 on the trail.

This benchmark problem is still repeatedly used (Georgiou and Teahan, 2010; Lehman and Stanley, 2010; Doucette and Heywood, 2010) because of its interesting characteristics: it has a large search space, full of global and local optima and many plateaus riven with deep valleys, which may be indicative of real problem spaces (Langdon and Poli, 1998). More, there are low and middle order schemas which are required as stepping stones to build solutions but which are below average fitness. Langdon and Poli (1998) have shown that GP does not perform significantly better than random search on this problem since it contains multiple levels of deception, where solutions have low fitness neighbors. The search space is highly deceptive because it has many high scoring non-optimal programs that do not correspond to trail following behavior: random policies may collect many food items.

### Performance of GE on the Santa Fe Trail Problem

O'Neill and Ryan (2003) compared Grammatical Evolution with Genetic Programming, using the Santa Fe Trail problem as a benchmark, fixing the steps upper limit to 615. They have used the grammar in Fig. 4 (from now on called BNF-O'Neill grammar) and the experiments results were fa-

```
<code> ::= <line> | <code> <line>
<line> ::= <condition> | <op>
<condition> ::= ifelse food-ahead
               [ <line> ] [ <line> ]
<op> ::= turn-left | turn-right | move
```

Figure 4: "BNF-O'Neill grammar definition.

```
<expr> ::= <line> | <expr> <line>
<line> ::= ifelse food-ahead
               [ <expr> ] [ <expr> ] | <op>
<op> ::= turn-left | turn-right | move
```

Figure 5: "BNF-Koza grammar definition.

vorable to GE.

The comparison SFT experiments were later questioned by Robilliard et al. (2005). They have found that the search space of programs possible by BNF-O'Neill grammar was not semantically equivalent to the set of programs possible within the original SFT definition. The former is narrower: any original SFT program can be translated into a semantically equivalent BNF-O'Neill program but the converse is not true. Robilliard et al. (2005) argue that they found no solution in the Santa Fe Trail problem using Grammatical Evolution, up to and including 600 time steps in their experiments, and they note that almost all Grammatical Evolution publications mention an upper limit of 615 time steps. Only in O'Neill and Ryan (2001) with Grammatical Evolution is the limit reported as 600 steps, which Robilliard et al. (2005) claim is a mistype.

Georgiou and Teahan (2010) have made a series of experiments where GE gives very poor results in the Santa Fe Trail problem with a search space semantically equivalent to the search space used in the original problem (Koza, 1991). They have used the SFT-BAP grammar cited in Robilliard et al. (2005), from now on named BNF-Koza (see Fig. 5), which defines a search space of programs semantically equivalent to the Koza's original (Robilliard et al., 2005). Indeed, the same work proved experimentally that GE literature (O'Neill and Ryan, 2001, 2003) uses a BNF grammar which narrows the original search space and consequently gives an unfair advantage to GE when it is compared against GP in the SFT problem. As noted by Robilliard et al. (2005) the BNF-O'Neill grammar does not allow multiple <op> statements or sequences of <op> and <condition> statements in the branches of the <condition> production rule, in contrast with the first production rule of <line> in the BNF-Koza.

In the experiments of Georgiou and Teahan (2010) the maximum allowable steps for the ant was set to 650. The reason for this increase was to give Grammatical Evolution the chance to find more solutions using the investigated BNF grammars in order that the sample of the solutions found

and the comparison of the effects of these grammars on the performance of Grammatical Evolution being more encompassing. Furthermore, they have proved experimentally that during an evolutionary run GE is not capable of finding solutions using the BNF-O'Neill grammar definition with less than 607 steps. BNF-O'Neill defines a smaller search space, biasing the search towards areas where a solution may be found more easily (higher success rate) but with the cost of excluding areas where more efficient programs (using less steps) exist. Note that the easy incorporation of domain knowledge, by changing the grammar, and biasing the search, is a great advantage of the grammar representation used by GE. But we should be careful not to exclude good solutions (the ones which require less steps), as it is the case of BNF-O'Neill.

The best solution we know so far for the SFT problem, in GP or GE, requires only 337 steps, and it was generated by CGE (Georgiou and Teahan, 2011), one of the GE variations developed over the last years.

### Promoting diversity in GP to overcome deception

Diversity promotion and maintenance, relying generally on the genotype and fitness function, has been the main approach to mitigate deception and stagnation in GP. A variety of different techniques have been proposed in order to preserve genotypic diversity, that make use of different measures to compare genotypes. We are going to refer only some of them: higher mutation rates (Banzhaf et al., 1996), larger population sizes (Ciesielski and Mawhinney, 2002), fitness shaping (Luke, 1998), aiding explicit objectives to promote genotypic diversity, together with a goal-oriented one, in a multi-objective scheme (de Jong et al., 2001), replacing the most similar programs (Ciesielski and Mawhinney, 2002), maintaining a diversity of genetic lineages (Burke et al., 2003) and fitness sharing (Ekárt and Németh, 2002). Burke et al. (2002) have shown that genotype diversity approaches manifest a low correlation with fitness.

A different approach is the promotion of phenotypic diversity based on fitness: the use of a selection method which is uniform over the fitness values (Legg and Hutter, 2005) or the phenotype diversity measurement in terms of the number of unique fitness values in the population, using entropy (Rosca, 1995). Burke et al. (2002) concluded that "successful evolution occurs when population converges to similar structure and high fitness diversity" and "the fitness based measures of phenotypes and entropy appear to correlate better with run performance." The phenotype techniques spread individuals across different fitness levels but are not able to maintain diversity in the same fitness levels (Legg and Hutter, 2005). Yan and Clack (2009) pointed out the importance of preserving phenotypic behavioral diversity, in achieving higher fitness and adaptability to dynamic domains, where behavior is not reduced to fitness, being much more detailed.

### Novelty Search

All diversity preserving techniques, described above, take objective-based fitness into account. The Novelty Search (NS) approach, introduced by Lehman and Stanley (2008) is much more radical, as it ignores completely the fitness objective, relying only on behavior diversity as the sole criteria for selection in artificial evolution. Objective based fitness is replaced by novelty and the idea is not to select the most fitted individuals for reproduction but those with the most novel behaviors instead. Novel individuals are rewarded and will guide the search towards finding other novel individuals. NS is driven towards behavior space exploration, looking for what is divergent from the past and present behaviors, regardless of their fitness. The idea is that by exploring the behavior space without any goal besides novelty, ultimately an individual with the desired behavior will be found. Surprisingly, NS has been successful in several deceptive problems (Lehman and Stanley, 2011; Gomes et al., 2012), as it is not dependent of any fixed goal, avoiding the convergence towards local optima.

NS requires the definition of distances between behavior descriptors. Those descriptors may be specific to a task or suited for a class of tasks (Doncieux and Mouret, 2010). The descriptors are normally vectors that capture behavior information along the whole evaluation or which is just sampled at some particular instants. The used descriptors may even change along an evolutionary run. Several general distance functions between behavioral vectors have been suggested: euclidian distance, edit distance, hamming distance, relative entropy and normalized compression distance, fourier analysis, for instance (Doncieux and Mouret, 2010).

Given a behavior function and a distance metric, the novelty score of an individual is computed as the average distance from its  $k$ -nearest neighbors ( $\mu_i$ ) in both the population and the archive (see Eq. 1).

$$\rho(x) = \frac{1}{k} \times \sum_{i=0}^k \text{dist}(x, \mu_i) \quad (1)$$

A point in a sparse area will be highly rewarded and a point in a dense area will receive a low novelty score. There have been several proposed ways of registering the past behaviors but we won't deal with them here, as in our NS application we won't use any archive of past behaviors.

### Novelty Search applied to GP

Lehman and Stanley (2010) were the first to apply novelty search to Genetic Programming. They made experiments in three deceptive tasks: maze navigation, and the two artificial ant benchmarks: the Santa Fe Trail and Los Altos. There is no reference to the features of the best individual evolved, in terms of the number of steps, genotype or phenotype length, but in the end, NS was able to avoid bloat, evolving smaller

programs and outperforming objective based fitness in terms of the number of successfully evolutionary runs.

Doucette and Heywood (2010) used SFT to study the effects of NS on performance and also on solution generalization. NS evolved programs, which eat less food items on average than the traditional fitness based method but they achieved better performance in terms of generalization to new trails.

Naredo et al. (2013) applied NS to evolve GP classifiers and obtained encouraging results when compared to canonical GP. NS exhibited the best results when confronted with difficult problems, but for simple problems, however, the explorative capacity of NS seemed to be a detriment to the search.

### Santa Fe Trail Experiments

All experiments mentioned in this study have been performed using the jGE library (Georgiou and Teahan, 2006), which is a Java implementation of the Grammatical Evolution algorithm, and jGE Netlogo extension (Georgiou and Teahan, 2010), which is a Netlogo extension of the jGE library. We have used the customary fitness function: the number of food items eaten. An invalid individual has a fitness of 0.

In order to evaluate the performance of Novelty Search in the Santa Fe Trail benchmark, we have used both grammars, BNF-ONeill and BNF-Koza. The latter is used because it is semantically equivalent to the original formulation of SFT, and we have used BNF-ONeill also because it has been extensively used in GE literature, although it narrows the original search space excluding good solutions. In order to sample a larger number of solutions we have fixed the maximum allowable steps for the ant to 650, as in (Georgiou and Teahan, 2010).

Note that when Grammatical Evolution is being performed with Novelty Search (NS-GE) every individual must be evaluated to know its fitness score. But that score is not being used for individual selection. The fitness score is only used to evaluate NS-GE performance.

We did not add any archive, thus novelty was only tested against the other individuals in the current population. Our preliminary experiments showed that the addition of an archive of past behaviors did not improve the performance of NS without memory. Moreover, the archive would introduce a memory mechanism that does not exist in the standard GE. In order to track novelty GE requires a small change: the fitness function is replaced by the novelty metric.

### Behavior descriptors

We have evaluated NS using several behavior descriptors, and all of them were compared against the standard GE for both grammars. There is no natural behavior descriptor: we have many possibilities and we wanted to make empirical

comparisons between several of them. The behavior descriptors should ideally be compact in order to capture relevant behavior variation and condense the irrelevant ones. We have to be careful not to conflate the stepping-stones that will lead the evolution towards finding successful solutions. We are going to present our four behavior descriptors (two of them are different samplings of the same behavior characteristic):

**Amount of food eaten** This is the behavior descriptor used by Lehman and Stanley (2010): the amount of food eaten is sampled  $N$  evenly spaced times during an individual evaluation. We have considered two values for  $N$ , in order to cover descriptors with different granularity and implying behavior spaces with different sizes:  $N=1$  and  $N=26$ . In the former case, the amount of food is sampled only once: at the end of the 650 steps or when the food trail is depleted. A sample of 1 means a behavior space of size 90 and a lot of conflation: many different forms of eating the same amount of food will be considered identical. A sampling of 26 means that the amount of food will be registered every 25 steps. The search space of behaviors will be much larger but there are constraints: it will be impossible to eat more than 25 items in each period. An invalid individual or a valid individual that does not eat any food after 650 steps will have a vector filled with 26 zeros or 1 zero depending on the sampling size. We have used the Euclidean distance for measuring similarity between behaviors.

**Food eaten sequence** This is a behavior descriptor similar to the one used by Doucette and Heywood (2010) in his SFT experiments. The behavior descriptor is a vector of size  $89 \times 2$ , containing the food item coordinates in the order in which they were eaten. The  $x$  and  $y$  coordinates of the first food item eaten occupy the first and second vector position, respectively. The 3rd and 4th vector positions are occupied respectively by the  $x$  and  $y$  coordinates of the second food item eaten. If the ant ate  $N$  food items, being  $N$  less than 89, the first  $N \times 2$  vector cells will be occupied by the  $N$  food item position coordinates and all the other vector cells will be filled by the  $x$  and  $y$  coordinates of the  $N$ th food item position. We have to describe the behavior of ants that do not eat any food item and also the invalid individuals. Both will have a vector filled with 0s. Note that not all of the food position combinations are possible because there are natural constraints imposed on the space of behaviors by the number of maximal steps. For example, it will be impossible to have the food items in reverse order: going to the end of the trail and then following it upside down. We will use the Euclidean distance for measuring similarity between behaviors.

**Steps sequence** We have a vector of 89 cells where the  $n$ th cell will be filled by the number of steps necessary to eat the  $n$ th food item. In case the ant has eaten only  $N$  ( $<89$ ) food items, every cell after the  $N$ th will be filled by the dummy

value. For example, this vector  $[1, 2, 3, *, *, *]$  means that one food item was eaten after the first move, a second food item after the second move, a third food item after three steps and then no more food items were eaten. An invalid individual will have a vector filled with the value \*. Note that again the behavior descriptor imposes constraints on the space of possible behaviors. The vector is always sorted in increasing order, except the dummy \*. We have used a variation on the Euclidean distance (see Eq. 2) for measuring the distance between behavior vectors using this descriptor. The subtraction in the Euclidean formula is substituted by a function  $dif(v, w)$  in order to deal with the dummy value \* (absence of food). Consider two vectors  $v$  and  $w$ , each one with 89 cells:  $v = (v_1, v_2, \dots, v_{89})$  and  $w = (w_1, w_2, \dots, w_{89})$ .

$$dist(v, w) = \sqrt{\sum_{i=0}^{89} dif(v_i, w_i)^2} \quad (2)$$

We want to penalize, as being different, the behavior vectors that have less food items (more \*s). This way the difference between a dummy value and any number of steps is 500. The difference function between the values from two corresponding cells is given by Eq. 3.

$$dif(x, y) = \begin{cases} 0 & \text{if } x=* \text{ and } y=* \\ x - y & \text{if } x \neq * \text{ and } y \neq * \\ 500 & \text{otherwise} \end{cases} \quad (3)$$

## Experiments setup

Five distinct experiments were performed to evaluate the standard GE and NS-GE with the four different descriptors using the BNF-Koza grammar and another five for BNF-ONeill. Each experiment consisted on 100 evolutionary runs. For each method, we have measured performance in terms of the success or hit rate (number of runs evolving individuals that eat all 89 food items) and the average quality of hits, (average number of steps necessary to eat all the 89 items). We also have measured the average number of food items eaten, the average number of generations for a hit, and finally the minimal number of steps of a hit. The tableau in Table 1 shows the settings and parameters of the experiments.

After some exploration, for almost all behavior descriptors we have chosen 10 for the parameter  $K$ , i.e. the number of nearest neighbors checked in NS (see Eq. 1). The only exception was the steps sequence descriptor: the best value for  $K$  was 5. We have used 50 as the maximum number of generations and a population of 500 individuals, which are widely used values in GP for the SFT. A Generation Gap of 90% means that in a population of 500 individuals the 50 best individuals will survive in the next generation and the worst 450 are replaced. Also, the standard genetic operators of mutation (point) and crossover (one-point) have been used.

|                        |                                                                                                                                                                                                                                                  |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Objective              | Find an ant that follows food trails                                                                                                                                                                                                             |
| Terminal Set           | turn-left, turn-right, move, food-ahead                                                                                                                                                                                                          |
| Behavior Descriptors   | Amount of food eaten sampled at the end - (K=10, Euclidean), Amount of food eaten sampled 26 times - (K=10, Euclidean), Food eaten sequence - (K=10, Euclidean), Steps sequence - (K=5, A specific Euclidean distance adapted to the descriptor) |
| Grammars               | BNF-O'Neill and BNF-Koza                                                                                                                                                                                                                         |
| Evolutionary Algorithm | Steady-State GA, Generation Gap: 0.9, Selection Mechanism: Roulette-Wheel                                                                                                                                                                        |
| Initial Population     | Randomly created with the following restrictions: Minimum Codons: 15 and Maximum Codons: 25                                                                                                                                                      |
| Parameters             | Population Size: 500, Maximum Generations: 50 (without counting generation 0), Prob. Mutation: 0.01, Prob. Crossover: 0.9, Codon Size: 8, Wraps Limit: 10                                                                                        |

Table 1: Summarizing Tableau for the SFT Problem.

## Results

The experimental results for BNF-Koza and BNF-O'Neill can be seen in Table 2 and Table 3, respectively. They show that GE with NS, without any memory of past individuals, using any of the four behavioral descriptors, clearly outperforms standard fitness based GE in terms of hits rate for BNF-Koza, which is semantically equivalent to the original SFT problem. Standard GE for BNF-O'Neill, which defines a biased narrower search space, had a high hit percentage but was also outperformed by NS-GE for every behavior descriptor. NS-GE was also able to find more efficient programs (requiring less number of steps) than standard GE in both grammars. Anyhow, in spite that NS-GE with Steps Sequence descriptor found the most efficient solution, we cannot say that any of those descriptor is more efficient than the other: chance can play a role here, on average the number of required steps is very similar. Successful solutions evolved by NS-GE required a less number of evaluations, for both grammars (confirmed by the values on the Gen column in both tables). The average number of food items eaten is much higher in the NS-GE for every descriptor but that comes naturally from the higher success rate.

Even a simple descriptor that defines a very small behavior space (90 behaviors), and which is highly related with the fitness function: just counting the amount of food eaten in the end of the 650 steps, was able to attain a remarkable success rate. When we increase the number of samples (granularity of behavior characterization), we have increased the behavior space but we were also able to distinguish behaviors that were considered similar with just one sampling. Note that the amount of food eaten can never decrease and



| BNF-Koza           | Hits | Food  | Gen   | Steps  | MinSteps |
|--------------------|------|-------|-------|--------|----------|
| Fitness (GE)       | 8%   | 58.8  | 37.75 | 589.25 | 497      |
| Food <sub>1</sub>  | 42%  | 78.36 | 22.55 | 599.38 | 385      |
| Food <sub>26</sub> | 48%  | 81.96 | 28.38 | 584.90 | 461      |
| Food Seq.          | 33%  | 76.59 | 30.28 | 575.10 | 385      |
| Steps Seq.         | 41%  | 77.88 | 23.78 | 587.92 | 331      |

| BNF-O'Neill        | Hits | Food  | Gen   | Steps  | MinSteps |
|--------------------|------|-------|-------|--------|----------|
| Fitness (GE)       | 77%  | 82.92 | 15.84 | 613.99 | 609      |
| Food <sub>1</sub>  | 95%  | 88.28 | 08.96 | 614.01 | 607      |
| Food <sub>26</sub> | 100% | 89.00 | 08.19 | 614.06 | 607      |
| Food Seq.          | 94%  | 88.24 | 09.64 | 613.87 | 607      |
| Steps Seq.         | 95%  | 88.28 | 08.96 | 614.01 | 607      |

Table 2: Results for BNF-Koza and BNF-O'Neill. For each behavior descriptor and objective based fitness, the columns give the percentage of hits, the average number of food items eaten, the average number of generations for a hit, the average and minimal number of steps for a hit.

26 samples imply sampling every 25 steps, which in combination with the form of the Santa Fe Trail impose strong constraints on the possible values of consecutive cells. The increase on the behavior description granularity has increased the performance in both grammars, but more experiments need to be made in order to assess if a too granular description is not necessary, distinguishing a lot of irrelevant behavior and slowing down evolution.

Considering all evolved programs in our experiments, the evolved code presented in Fig. 6 was able to complete the trail in the least number of steps: 331. This program outperforms the most efficient program known so far, evolved by the Constituent Grammatical Evolution algorithm (CGE) (Georgiou and Teahan, 2011). Note that CGE has attained a higher hit rate than NS-GE, but CGE is a much more complex algorithm than the standard GE, incorporating the concepts of constituent genes and conditional behavior switching, being able to decrease the actual search space, and explore programs in more useful areas. With a slight modification over the standard GE, using very intuitive behavior characterizations, by rewarding individuals that behave differently from the others, without any selective pressure to find the better ones, NS-GE has improved dramatically the performance on the very deceptive Santa Fe Trail Problem.

## Conclusion and Future Work

Even though Grammatical Evolution shows competence on a series of problems where it has been applied, its effectiveness and efficiency on deceptive problems, such as the Santa Fe Trail, has been questioned (Robilliard et al., 2005). This is because problems of this type have some characteristics of real world problems, such as many local optima and large

```

ifelse food-ahead
[ move move ]
[ turn-left
  ifelse food-ahead
    [ move move ]
    [ turn-right ]
]
turn-right
ifelse food-ahead
[ move ]
[ turn-left ]
move
    
```

Figure 6: Best Program Evolved. Netlogo code of the best solution (331 steps) with the descriptor Steps Sequence.

search spaces, making them challenging and difficult for evolutionary algorithms to efficiently solve them (Langdon and Poli, 1998; O'Neill and Ryan, 2003). This work presented the first application of Novelty Search in Grammatical Evolution and benchmarked this approach in the Santa Fe Trail problem to investigate whether a search mechanism, which promotes behavioral novelty, would improve the performance of Grammatical Evolution in a deceptive problem. The results demonstrated a dramatic improvement in terms of both success rate and quality of solutions, which encourages further investigation and application to more problems such as Los Altos Hills and Maze Searching. Furthermore, more work is required for the investigation of how the application of Novelty Search in Grammatical Evolution affects the genotype and phenotype bloating, a known issue with the Santa Fe Trail problem.

## Acknowledgments

This work was supported by FCT strategic project PEst-OE/EEI/UI0434/2011.

## References

- Banzhaf, W., Francone, F. D., and Nordin, P. (1996). The effect of extensive use of the mutation operator on generalization in genetic programming using sparse data sets. In *Parallel Problem Solving from Nature IV, Proceedings of the International Conference on Evolutionary Computation*, edited by, pages 300–309. Springer Verlag.
- Burke, E. K., Gustafson, S., Kendall, G., Krasnogor, N., and Gustafson, E. K. B. S. (2003). Is increased diversity in genetic programming beneficial? an analysis of lineage selection. In *Congress on Evolutionary Computation*, pages 1398–1405. IEEE Press.
- Burke, E. K., Gustafson, S. M., Kendall, G., and Krasnogor, N. (2002). Advanced population diversity measures in genetic programming. In *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, PPSN VII, pages 341–350, London, UK, UK. Springer-Verlag.
- Ciesielski, V. and Mawhinney, D. (2002). Prevention of early convergence in genetic programming by replacement of similar

- programs. In Fogel, D. B., El-Sharkawi, M. A., Yao, X., Greenwood, G., Iba, H., Marrow, P., and Shackleton, M., editors, *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, pages 67–72. IEEE Press.
- de Jong, E. D., Watson, R. A., and Pollack, J. B. (2001). Reducing bloat and promoting diversity using multi-objective methods. In Spector, L., Goodman, E. D., Wu, A., Langdon, W. B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M. H., and Burke, E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 11–18, San Francisco, California, USA. Morgan Kaufmann.
- Dempsey, I., O'Neill, M., and Brabazon, A. (2006). Adaptive trading with grammatical evolution. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 9137–9142, Vancouver. IEEE Press.
- Doncieux, S. and Mouret, J.-B. (2010). Behavioral diversity measures for evolutionary robotics. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE.
- Doucette, J. and Heywood, M. I. (2010). Novelty-based fitness: an evaluation under the santa fe trail. In *Proceedings of the 13th European conference on Genetic Programming, EuroGP'10*, pages 50–61, Berlin, Heidelberg. Springer-Verlag.
- Ekárt, A. and Németh, S. Z. (2002). Maintaining the diversity of genetic programs. In Foster, J. A., Lutton, E., Miller, J., Ryan, C., and Tettamanzi, A. G. B., editors, *Genetic Programming, Proceedings of the 5th European Conference, EuroGP 2002*, volume 2278 of *LNCS*, pages 162–171, Kinsale, Ireland. Springer-Verlag.
- Georgiou, L. (2012). *Constituent Grammatical Evolution*. Phd thesis, School of Computer Science, Bangor University.
- Georgiou, L. and Teahan, W. J. (2006). jge: a java implementation of grammatical evolution. In *Proceedings of the 10th WSEAS international conference on Systems, ICS'06*, pages 410–415, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS).
- Georgiou, L. and Teahan, W. J. (2010). Grammatical evolution and the santa fe trail problem. In Filipe, J. and Kacprzyk, J., editors, *IJCCI (ICEC)*, pages 10–19. SciTePress.
- Georgiou, L. and Teahan, W. J. (2011). Constituent grammatical evolution. In Walsh, T., editor, *IJCAI*, pages 1261–1268. IJCAI/AAAI.
- Gomes, J. C., Urbano, P., and Christensen, A. L. (2012). Introducing novelty search in evolutionary swarm robotics. In Dorigo, M., Birattari, M., Blum, C., Christensen, A. L., Engelbrecht, A. P., Groß, R., and Stützle, T., editors, *ANTS*, volume 7461 of *Lecture Notes in Computer Science*, pages 85–96. Springer.
- Koza, J. R. (1991). Genetic evolution and co-evolution of computer programs. In Langton, C. T. C., Farmer, J. D., and Rassmussen, S., editors, *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, pages 603–629. Addison-Wesley, Santa Fe Institute, New Mexico, USA.
- Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, MA, USA.
- Langdon, W. B. and Poli, R. (1998). Why ants are hard. In Koza, J. R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M. H., Goldberg, D. E., Iba, H., and Rolo, R., editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 193–201, University of Wisconsin, Madison, Wisconsin, USA. Morgan Kaufmann.
- Legg, S. and Hutter, M. (2005). Fitness uniform deletion: a simple way to preserve diversity. In Beyer, H.-G. and Reilly, U.-M. O., editors, *Genetic and Evolutionary Computation Conference, GECCO 2005, Proceedings, Washington DC, USA, June 25-29, 2005*, pages 1271–1278. ACM.
- Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In *Proc. of the Eleventh Intl. Conf. on Artificial Life (ALIFE XI)*, Cambridge, MA. MIT Press.
- Lehman, J. and Stanley, K. O. (2010). Efficiently evolving programs through the search for novelty. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2010)*. ACM.
- Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223.
- Luke, S. (1998). Genetic programming produced competitive soccer softbot teams for robocup97. In *University of Wisconsin*, pages 214–222. Morgan Kaufmann.
- Naredo, E., Trujillo, L., and Martinez, Y. (2013). Searching for novel classifiers. In Krawiec, K., Moraglio, A., Hu, T., Uyar, A. S., and Hu, B., editors, *Proceedings of the 16th European Conference on Genetic Programming, EuroGP 2013*, volume 7831 of *LNCS*, pages 145–156, Vienna, Austria. Springer Verlag.
- O'Neill, M. and Ryan, C. (2001). Grammatical evolution. *IEEE Transactions on Evolutionary Computation*, 5(4):349–358.
- O'Neill, M. and Ryan, C. (2003). *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*. Kluwer Academic Publishers, Norwell, MA, USA.
- Robilliard, D., Mahler, S., Verhaghe, D., and Fonlupt, C. (2005). Santa fe trail hazards. In Talbi, E.-G., Liardet, P., Collet, P., Lutton, E., and Schoenauer, M., editors, *7th International Conference on Artificial Evolution EA 2005*, volume 3871 of *Lecture Notes in Computer Science*, pages 1–12, Lille, France. Springer. Revised Selected Papers.
- Rosca, J. (1995). Entropy-driven adaptive representation. In *Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications*, pages 23–32. Morgan Kaufmann.
- Ryan, C., Collins, J., Collins, J., and O'Neill, M. (1998). Grammatical evolution: Evolving programs for an arbitrary language. In *Lecture Notes in Computer Science 1391, Proceedings of the First European Workshop on Genetic Programming*, pages 83–95. Springer-Verlag.
- Yan, W. and Clack, C. D. (2009). Behavioural gp diversity for adaptive stock selection. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation, GECCO '09*, pages 1641–1648, New York, NY, USA. ACM.

# Quantifying the Impact of Parameter Tuning on Nature-Inspired Algorithms

Matthew Crossley, Andy Nisbet and Martyn Amos

School of Computing, Mathematics and Digital Technology,  
Manchester Metropolitan University, Manchester M1 5GD, UK.  
Email: m.crossley@mmu.ac.uk

## Abstract

The problem of *parameterization* is often central to the effective deployment of nature-inspired algorithms. However, finding the optimal set of parameter values for a combination of problem instance and solution method is highly challenging, and few concrete guidelines exist on how and when such *tuning* may be performed. Previous work tends to either focus on a specific algorithm or use benchmark problems, and both of these restrictions limit the applicability of any findings. Here, we examine a *number* of different algorithms, and study them in a “problem agnostic” fashion (i.e., one that is not tied to specific instances) by considering their performance on *fitness landscapes* with varying characteristics. Using this approach, we make a number of observations on which algorithms may (or may not) benefit from tuning, and in which specific circumstances.

## Introduction and Background

There exist many algorithms that are inspired by nature, and each has associated with it a set of *parameters*. These define specific features or details of an algorithm that may be altered in order to change the behaviour or performance of the method (for example, in evolutionary algorithms, parameters may include mutation rate or crossover probability). The problem of finding the optimal settings for these parameters (often referred to as “tuning”) is well-established (Lobo et al., 2007; Nannen et al., 2008; Akay and Karaboga, 2009; Birattari, 2009; Eiben and Smit, 2011), but little in-depth work has been performed on quantifying the *benefits* of tuning for a *range* of algorithms. We address this in the current paper, by investigating the precise benefits (or otherwise) of tuning for a number of different algorithms. Moreover, we do this in a way that is independent of any specific *problem*, by using an approach based on fitness landscape characteristics. The main contribution of the paper is therefore to establish a framework for deciding - prior to any problem-specific implementation - which algorithms may (or may *not*) benefit from tuning. Our aim is to offer advice to future practitioners on the relative merits of tuning, compared to the effort involved in finding the best set of parameter values. We achieve this by establishing, for each

algorithm, the problem features that offer the most *potential* for performance improvements via tuning.

Previous work (Crossley et al., 2013) characterised a number of nature-inspired algorithms according to their performance on fitness landscapes with different features. However, the authors used the *default* parameter settings for each algorithm, which fails to reflect the fact that, in practice, methods are usually *tuned* prior to serious use (Leung et al., 2003; Adenso-Diaz and Laguna, 2006; Koster and Beney, 2007; Ridge and Kudenko, 2010). Here, we extend this work by quantifying the relative merits of tuning for a range of algorithms in a wide variety of fitness landscape scenarios. We achieve this by assessing both their tuned and untuned behaviour, using the methods described in Crossley et al. (2013).

In order to tune our selected algorithms, we use the notion of *racing*, which was first introduced in the field of machine learning (Maron and Moore, 1993, 1997). Specifically, we use the F-race algorithm (Birattari et al., 2002; Yuan and Gallagher, 2004; Smit and Eiben, 2009; Birattari et al., 2010), which has been extensively used to find the best possible set of parameter values for a given problem in a limited time.

The rest of the paper is organised as follows: we first describe our approach in the Methodology section, before presenting our experimental findings in the Results section. We conclude with a discussion of the implications of our results, and suggest further work.

## Methodology

Our methodology may be summarised as follows: (1) select a number of nature-inspired algorithms, and obtain consistent source code for their implementation; (2) for each algorithm, find the best parameter settings (i.e., tune) over a number of different problems; (3) compare the performance of tuned and untuned algorithms.

### Algorithm selection

We compare a number of nature-inspired algorithms, all of which are commonly applied to continuous function opti-

misation (we use the same set as in Crossley et al. (2013)). These may be classified (Brabazon and O'Neill, 2006) as either *social*, *evolutionary* or *physical*. The social algorithms we select are Bacterial Foraging Optimisation Algorithm (BFOA) (Passino, 2002), Bees Algorithm (BA) (Pham et al., 2006), and Particle Swarm Optimisation (PSO) (Kennedy and Eberhart, 1995). The evolutionary algorithms selected are Genetic Algorithms (GA) (Goldberg, 1989) and Evolution Strategies (ES) (Bäck and Schwefel, 1993), and physical algorithms are represented by Harmony Search (HS) (Geem and Kim, 2001). We also include stochastic hill climbing (SHC) as a “baseline” algorithm; in contrast to Crossley et al. (2013) we exclude random search, as it has no parameters to tune. As before, we heed the observation that “Ideally, competing algorithms would be coded by the same expert programmer and run on the same test problems on the same computer configuration” (Barr et al., 1995). With that in mind, we use only implementations provided by Brownlee (2011). Space prevents a complete description of specific implementation details for each algorithm, but full implementation details can be found in Brownlee (2011), which is freely available and contains complete source code.

## Tuning

Our fundamental goal is to investigate the pre- and post-tuned performance of our selected algorithms on landscapes with different general features, and thus identify characteristics of landscapes for which tuning may yield significant differences in algorithm performance. As Morgan and Gallagher (2010) observe, “Different problem types have their own characteristics, however it is usually the case that complementary insights into algorithm behaviour result from conducting larger experimental studies using a variety of different problem *types*” (our emphasis). Rather than using arbitrary benchmark instances of problems in order to perform tuning, we use a landscape-based approach, as utilised in Crossley et al. (2013). As Morgan and Gallagher (2010) explain, this Max-Set of Gaussians (MSG) method (Gallagher and Yuan, 2006) is a “randomised landscape generator that specifies test problems as a weighted sum of Gaussian functions. By specifying the number of Gaussians and the mean and covariance parameters for each component, a variety of test landscape instances can be generated. The topological properties of the landscapes are intuitively related to (and vary smoothly with) the parameters of the generator.” We now describe the characteristics under study:

*Ruggedness* of a landscape is often linked to its difficulty (Jones and Forrest, 1995), and factors affecting this include (1) the *number* of local optima (Horn and Goldberg, 1994), and (2) *ratio* of the fitness value of local optima to the global optimal value (Malan and Engelbrecht, 2009; Merz, 2000). Other significant factors concern (3) *dimensionality* (Hendtlass, 2009) (that is, the number of variables in the objective function), (4) *boundary constraints* (that is, the limits im-

Table 1: A summary of the ranges selected for the various characteristics in the landscape generation methodology.

| Characteristic                          | Min | Max | Step | Default |
|-----------------------------------------|-----|-----|------|---------|
| No. of local optima                     | 0   | 9   | 1    | 3       |
| Ratio of local optima to global optimum | 0.1 | 0.9 | 0.2  | 0.5     |
| Dimensionality                          | 1   | 10  | 1    | 2       |
| Boundary constraints                    | 10  | 100 | 10   | 30      |
| Smoothness                              | 10  | 100 | 10   | 15      |

posed on the value of a variable) (Kukkonen and Lampinen, 2005), and (5) *smoothness* of each Gaussian curve (effectively the gradient) used to generate the landscape (Beyer and Schwefel, 2002) - a smaller value indicates a smoother gradient. For each characteristic, we use the same ranges as in Crossley et al. (2013), summarised in Table 1.

To produce a test set of problems, we use the MSG landscape generator. For every value of every characteristic (in the range specified in Table 1) we generate a set of five landscapes, which makes up the initial problem set for each value. We then use the F-racing methodology (Birattari et al., 2002) to find optimised parameters for each algorithm, over *every* value of *every* landscape characteristic used. We ensure that termination criteria are standardised, in order to ensure reasonable comparisons, and therefore use the number of objective function evaluations to determine when to terminate an algorithm’s run. We established, through initial experiments, that all selected algorithms generally converge within 20,000 objective function evaluations, so we use that as the specific value.

## Comparison

We run each algorithm 100 times on each landscape in the set of landscapes generated for each particular characteristic value (when investigating smoothness, for example, we generate 1000 different landscapes (100 each for smoothness = 10 ... 100), and run each algorithm 100 times on each landscape). This is done first for all algorithms with ‘default’ parameter configurations, and then again, this time using the parameter configurations obtained through the F-Racing process. We measure the performance of each algorithm in terms of the mean ( $\mu$ ) and standard deviation ( $\sigma$ ) of the exact average error obtained, over all values for a particular characteristic. That is, we investigate the *robustness* of each algorithm to changes in the values for each characteristic, rather than their absolute performance on specific problem instances. This allows us to identify specific landscape features where tuning may make a significant difference, some difference, or no difference at all, for a particular algorithm.



## Results

We find that the effect of tuning using F-Racing is varied across algorithms, and that they fit into three categories: Algorithms which *do not* benefit from F-Racing (ES), algorithms which only benefit significantly from F-Racing when a landscape is ‘difficult’ for the algorithm using default parameters (BA, HS, PSO), and algorithms which *always* benefit from F-Racing (BFOA, GA, SHC). Of course, we acknowledge the fact that F-Racing is just one of many possible meta-search techniques available for parameter tuning, and future work will involve a comparative study of alternative methods.

We summarise our results in Table 2; the full datasets are available online<sup>1</sup>; the repository contains all performance data across all runs, summary spreadsheets and details of all parameter settings. We now examine in detail the performance of each algorithm, using spider plots to graphically depict the results in Table 2. For each plot, the further a line is from the origin, the *smaller* the average error (that is, the “larger” an area, the larger the degree of robustness, which is considered “better”).

### Bacterial Foraging Optimisation Algorithm

There exists little discussion on the role of different parameters in the BFOA. While some elements of the search pattern are clearly altered by various parameters, it is very difficult to estimate values for these. In the original description of the BFOA (Passino, 2002), the parameter values were assigned based on observation of actual bacterial colonies. While this may be true to the nature-inspired concept, it is not necessarily the best way to obtain optimal performance from the algorithm. The combination of parameters offered by BFOA gives a highly configurable search environment. Parameters such as step size and population size directly affect the potential area the algorithm can explore in a given number of objective function calculations. Attraction and repulsion weights, and the “space” over which these attraction and repulsion effects spread, work to control local optima avoidance. Parameters controlling the number of chemotactic steps before a reproduction step, and the number of reproduction steps before an elimination-dispersal event, control the balance of *exploitation* versus *exploration*. Given that the search behaviour of the algorithm is highly configurable, it is unsurprising that BFOA is heavily reliant on tuning. Results for BFOA are shown in Figure 1. Across all characteristics, tuning offers a *significant* improvement on the average error and standard deviation of the performance - in many cases, improving the ranking of the algorithm from the largest average error to one of the smallest, and coping well with the changing characteristics. We see the most significant improvement where boundary constraint ranges change, a characteristic that is heavily reliant on parameters

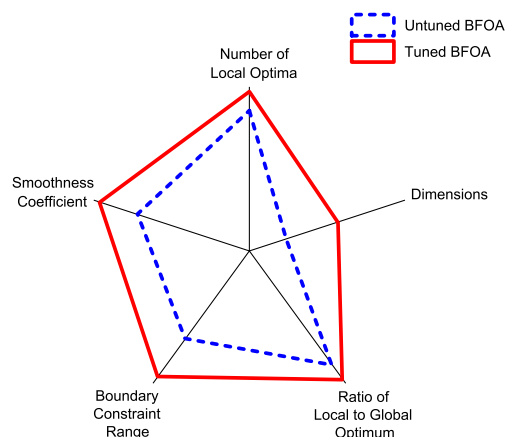


Figure 1: Summary of results for **Bacterial Foraging Optimisation Algorithm**.

which control the range at which new solutions are generated (in the case of BFOA, this is the *step size*). Improvements are also shown for dimensionality and smoothness coefficient, increasing the performance of BFOA where there is little gradient information in a large fitness landscape. Smaller improvements are demonstrated by the increasing number of local optima and the increasing attractiveness of these local optima, but tuning still benefits the algorithm considerably.

In terms of the configurations selected by F-Racing, there is little variation in parameter values as characteristics change. Across all characteristics, and all values for those characteristics, there are only eight different configurations selected by racing. This suggests that, while it is difficult to find a *good* configuration, once it has been found, it is likely to be good for all *similar* problems. Tuning is vital to the performance of the BFOA, but it is possible that by exploring problems using a similar methodology to that demonstrated here, we may create a ‘bank’ of promising configurations.

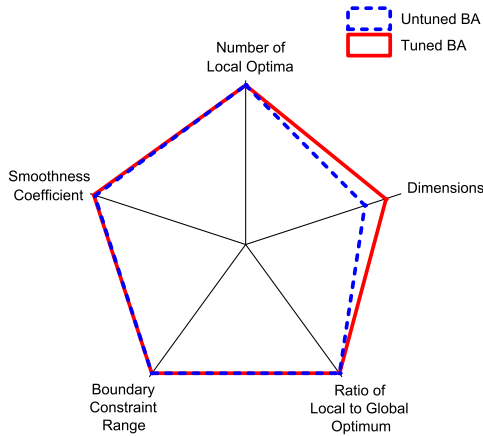
### Bees Algorithm

The BA is considered to be an algorithm on which parameterisation has little effect (Pham et al., 2006). We observe that the BA is one of the best *untuned* performers in this study, offering weight to this argument for relative parameter insensitivity. In terms of adjusting the BA to cope with an increasing number of local optima, there are several parameters which have an effect. Parameters such as the *number of sites* under investigation, the *number of bees* attributed to those sites, and the *differentiation* between sites and ‘elite’ sites are all factors which affect the searching behaviour of the algorithm to allow for greater flexibility as the modality of the problem landscape increases. Results for BA are

<sup>1</sup><http://dx.doi.org/10.6084/m9.figshare.696908>

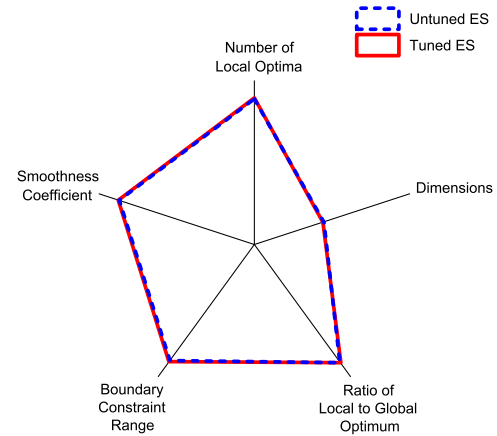
Table 2: Mean ( $\mu$ ) and standard deviation ( $\sigma$ ) of the exact average error of algorithm performance (both untuned (UT) and tuned (T)). Smaller values imply more robustness to changes in a specific characteristic.

|                    |          | BFOA  |       | Bees Algorithm       |                      | ES    |       | GA    |       | Harmony Search |                      | PSO   |       | SHC   |       |
|--------------------|----------|-------|-------|----------------------|----------------------|-------|-------|-------|-------|----------------|----------------------|-------|-------|-------|-------|
| # of Local Optima  | $\mu$    | UT    | T     | UT                   | T                    | UT    | T     | UT    | T     | UT             | T                    | UT    | T     | UT    | T     |
|                    | $\sigma$ | 0.118 | 0.003 | 0.001                | $8.8 \times 10^{-6}$ | 0.085 | 0.078 | 0.093 | 0.015 | 0.011          | $2.2 \times 10^{-5}$ | 0.025 | 0.014 | 0.266 | 0.072 |
| Dimensions         | $\mu$    | 0.011 | 0.001 | $2.1 \times 10^{-4}$ | $9.2 \times 10^{-7}$ | 0.028 | 0.026 | 0.033 | 0.008 | 0.005          | $2.9 \times 10^{-5}$ | 0.010 | 0.010 | 0.041 | 0.020 |
|                    | $\sigma$ | 0.754 | 0.417 | 0.216                | 0.073                | 0.542 | 0.544 | 0.420 | 0.529 | 0.364          | 0.263                | 0.420 | 0.157 | 0.577 | 0.589 |
| Local Optima Ratio | $\mu$    | 0.388 | 0.360 | 0.202                | 0.069                | 0.345 | 0.346 | 0.233 | 0.401 | 0.271          | 0.204                | 0.307 | 0.145 | 0.261 | 0.371 |
|                    | $\sigma$ | 0.120 | 0.003 | 0.001                | $8.7 \times 10^{-5}$ | 0.084 | 0.082 | 0.079 | 0.007 | 0.007          | 0.002                | 0.025 | 0.016 | 0.284 | 0.088 |
| Boundary Range     | $\mu$    | 0.021 | 0.002 | $2.3 \times 10^{-4}$ | $1.9 \times 10^{-4}$ | 0.012 | 0.012 | 0.006 | 0.006 | 0.003          | 0.004                | 0.004 | 0.011 | 0.045 | 0.011 |
|                    | $\sigma$ | 0.317 | 0.022 | 0.001                | 0.001                | 0.097 | 0.093 | 0.125 | 0.021 | 0.048          | 0.001                | 0.076 | 0.022 | 0.446 | 0.305 |
| Smoothness         | $\mu$    | 0.213 | 0.033 | $1.3 \times 10^{-4}$ | 0.001                | 0.017 | 0.018 | 0.057 | 0.016 | 0.041          | 0.001                | 0.050 | 0.013 | 0.239 | 0.217 |
|                    | $\sigma$ | 0.260 | 0.010 | 0.004                | 0.001                | 0.110 | 0.102 | 0.154 | 0.021 | 0.018          | 0.001                | 0.043 | 0.014 | 0.349 | 0.112 |
|                    |          | 0.089 | 0.005 | 0.002                | $4.3 \times 10^{-4}$ | 0.012 | 0.012 | 0.045 | 0.011 | 0.007          | 0.001                | 0.012 | 0.006 | 0.039 | 0.014 |


 Figure 2: Summary of results for **Bees Algorithm**.

shown in Figure 2. Post-tuning, we find that the BA selects the *same* parameter configuration, regardless of the number of local optima present in the landscape. We then see that tuning has no effect on the ability of the algorithm to cope with increasing numbers of local optima. As long as the number of sites under investigation is greater than the number of optima, the algorithm is capable of dealing with modality. Coupled with the abandonment of ‘unpromising’ sites, this means that ‘too many’ sites are not detrimental to the exploration pattern of the algorithm.

We see the same pattern when increasing the ratio of local optima to the global optimum. As long as the number of sites under investigation covers the modality of the landscape, the BA is not hampered by increasing levels of attractiveness, regardless of parameter settings. The *patch size* parameter of the BA controls the distance from a site bees are allowed to explore. This is the parameter which affects the search behaviour of the algorithm as boundary constraint size increases. The BA allows for full coverage of any sized search space, using *scout bees* to investigate new random sites to give ‘teleportation’ across the landscape. As with the number of local optima, we find the F-Races for the BA


 Figure 3: Summary of results for **Evolution Strategies**.

select the same parameter set for most boundary constraint sizes. We find that, post-tuning, the performance of the BA actually *decreases* slightly, suggesting the algorithm can cope less well with changes in boundary constraint size. We believe that the configurations may have become over-fitted to the landscapes used for tuning, and, while performance on the landscapes used for racing may have increased, the ability to search generalised landscapes has decreased. Dimensionality provides the most significant result in terms of pre-tuning and post-tuning performance of the BA. We observe little change in performance at one to three dimensions - the point where the untuned algorithm is already performing well. As dimensionality increases beyond this, the effect of tuning becomes *increasingly* beneficial. We suggest that there is no increase in performance in other characteristics because these landscapes are *not challenging enough* to the BA to require adjusting the parameters. For the ranges of landscape characteristics on which we have tested the BA, it is clear that tuning generally makes little difference to the performance, as suggested by its original developer.

## Evolution Strategies

ES has the smallest number of parameters of all the algorithms studied here (excepting the baseline algorithm, stochastic hill climbing). The two parameters this form of ES offers are (1) *population size* and (2) *number of children*. It is suggested (Cant-Paz, 2001) that altering these parameters adjusts *selection pressure* (that is to say, the *greediness* of the algorithm changes). The parameter configurations obtained through F-Racing are varied, implying that there do exist some configurations that are more successful than others. A range of configurations are selected across each characteristic - both in terms of different values for the two parameters, and different selection pressures when the two parameters are combined. Results for ES are shown in Figure 3. It is perhaps surprising to observe that the results of using the tuned parameters show little or no change in performance across all characteristics. There is a small decrease in average error as the number of local optima changes, but the standard deviation is similar for both untuned and tuned, suggesting that while the average error has decreased very slightly, the ability of the algorithm to cope with increasing numbers of local optima is unchanged. For all other characteristics post-tuning, there is little change in both average error and standard deviation across characteristics values (that is to say, the algorithm is no more capable of dealing with changes in these characteristics). This is perhaps consistent with the definition of the two parameters the algorithm offers - selection pressure can only affect the way in which ES explores local optima, and there is no control over the area that is explored around each point of interest, or any way to encourage the algorithm to rapidly explore an increasingly large search space.

We use a simple variant of ES, here, and there exist many other versions of the ES algorithm that offer a greater range of parameters (such as CMA-ES (Hansen and Kern, 2004)). ES clearly yields its best performance with an “out-of-the-box” parameter configuration, which means that it is quick to implement. However, our results suggest that there is little that can be done to improve the performance of this particular variant.

## Genetic Algorithm

The performance of the GA increases post-tuning, coping significantly better with increasing numbers of local optima, increasing boundary constraint range and an increasing smoothness coefficient. Results for the GA are shown in Figure 4. The parameters of the GA are not as intuitively linked to the exploration pattern as many of the other algorithms in the study. This particular GA offers four configurable parameters: (1) *population size*, (2) *‘bits’ per parameter* in the representation, (3) *crossover rate* and (4) *mutation rate*. In experiments with a fixed number of objective function calculations, population size affects the number of generations the algorithm evaluates before terminating. A

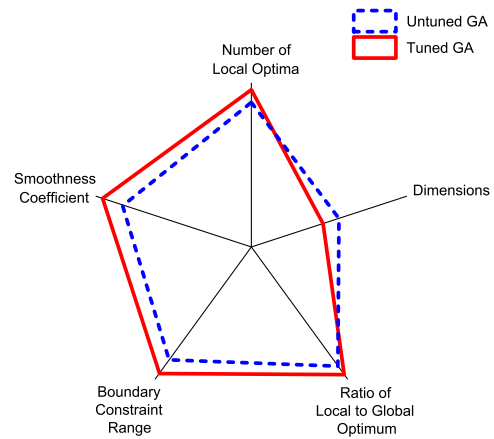


Figure 4: Summary of results for **Genetic Algorithm**.

larger number of bits in a bit string representation allows more ‘precise’ solutions to be generated at the expense of having a representation which is less affected by mutation. Similarly to BFOA, there are a few configurations which re-occur across different characteristics and different characteristic values. It is probable that once a ‘good’ configuration has been found for a GA, it is applicable to ‘similar’ landscapes, which is consistent with the suggestion Goldberg (1989) that GAs are robust problem solvers, exhibiting approximately the same performance across a wide range of problems.

With increasing dimensionality, the GA initially shows promising results in terms of tuned performance, with a marked performance increase up to four dimensions. The benefit from tuning rapidly declines, however, until the tuned performance is *worse* than that of the untuned version. There are two possible explanations for this: the first is that the restriction on the number of objective calculations did not allow the F-Race algorithm to gather any meaningful performance data from the configurations. The second explanation is that we did not test a wide enough range of configurations - although two of the four parameters have definite ranges (mutation and crossover rates are percentages, thus generation was bounded between zero and one), so this is unlikely.

## Harmony Search

The four parameters of HS all control different aspects of the search strategy. *Memory size* dictates how many promising solutions can be stored - effectively, how many potential sites of interest are retained by the algorithm. *Consideration rate* and *adjustment rate* control how new solutions are generated. The consideration rate is the percentage chance that a solution based on one in memory will be generated (conversely, *1-consideration rate* is the chance a random so-

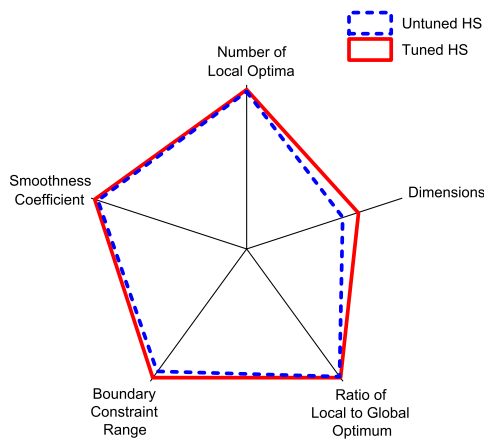


Figure 5: Summary of results for **Harmony Search**.

lution is generated instead). The adjustment rate is then the percentage chance that the randomly chosen solution from memory will be adjusted. If so, the fourth parameter, which controls the *maximum range* at which solutions can be adjusted, is used. If the adjustment does not occur, the considered solution potentially occupies an additional slot in the memory - thus increasing the chance that this solution may be chosen for consideration again. The interplay between these parameters is crucial, and it is somewhat hard to see how consideration rate and adjustment rate can directly affect the search strategy - unlike memory size and range, which are more obvious. The results for HS are shown in Figure 5. HS, like the BA, offers some of the lowest ‘out of the box’ average error rates in this study. For most characteristics, there is little room for a performance increase post-tuning. Boundary constraint range proves to be the second-most challenging characteristic to HS pre-tuning, but post-tuning shows improved performance. The range values in all the configurations selected by F-Racing are much smaller than those in the ‘out of the box’ values, and this contributes significantly to the performance improvement when boundary constraint ranges are increasing. The consideration rate also decreases almost linearly as size increases - effectively, more random solutions are used instead of relying on the ‘memory’. These random solutions allow the solution pool to jump from one position in the search space to another, encouraging a wider search space, and explaining the significant improvement as boundary constraint range increases. Dimensionality also yields an improvement in the tuned parameter performance of HS, in terms of both average error and ability to cope, as it rises. High dimension problems (seven and above) have a much higher consideration rate than the successful configurations for lower dimensionality, suggesting that a focus on *exploitation* rather than *exploration* is beneficial to the HS when dimensionality is high.

This is the opposite case of what happens with boundary constraint range, as discussed above.

## Particle Swarm Optimisation

PSO in this form has four parameters; these control the *population size*, the *maximum velocity* of a particle, the bias towards the *particle best solution* and the bias towards the *global best solution*. With these parameters, it is possible to control the coverage of a search space (the number of particles), enforce a large search area of a small search area for each particle (the maximum velocity), and, through manipulation of the local and global best solution bias, control the capability of the algorithm to converge on a single solution or explore several areas of interest (optima avoidance). Results for PSO are shown in Figure 6. The parameters used cover a broad range of search behaviours, and, as such, we would expect to see a large improvement in particle swarm performance post-tuning. This holds true for most of our characteristics. Results for the number of local optima, for example, show a reasonable decrease in average error as the number of local optima increases, yet the standard deviation demonstrates no change, indicating that the algorithm is no more capable of dealing with increasing numbers of local optima post-tuning. Performance of PSO greatly improves on dimensionality post-tuning, in terms of both average error and ability to cope as it grows. The F-Race algorithm for PSO selects the same configuration for all values of dimensionality (except for 2 dimensions), implying that there is no specific parameter that requires adjustment to cope with the increase in dimensionality, but selecting a configuration which provides good *exploration* allows PSO to perform well as the size of the search space increases exponentially. This trend continues across all characteristics, with F-Races often selecting the same configurations, regardless of characteristic values. As with the other swarming algorithms, we suggest that once a good configuration has been found, it is able to deal with a wide range of problems of a similar nature, regardless of the specific characteristics. The configurations selected are all varied in their parameters, and it is unexpected to see that there is no pattern to maximum velocity as boundary constraint range increases. This is possibly because maximum velocity is an upper bound, and there are particles with randomly generated velocities below the maximum, so this parameter is less significant than it may initially appear. It would perhaps be interesting to consider the effect of having a *minimum* velocity on the increase in boundary constraint range, although this would also severely hamper exploitation.

## Stochastic Hill-Climbing

With only a single parameter - the *range* at which new solutions are generated - the SHC algorithm does not offer a large amount of customisation. This single parameter is directly linked to the search pattern and nothing else, and as



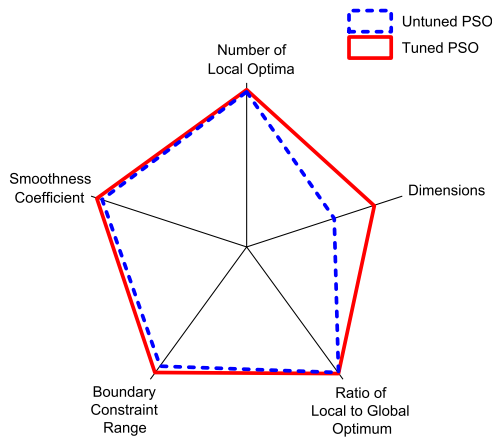


Figure 6: Summary of results for **Particle Swarm Optimization**.

there are no other parameters there is no interplay between parameters to consider. Arguably, therefore, SHC should prove the easiest algorithm to tune. Results for SHC are shown in Figure 7. All characteristics, barring dimensionality, show an improvement post-tuning. As the neighbourhood size is the range at which new solutions are generated, it is unsurprising that tuning improves algorithm performance as boundary constraint ranges change. As the number of objective function calculations is limited, despite having a larger neighbourhood size, the ability of the algorithm to effectively explore larger environments is still restricted, therefore the average error does not decrease by as much as may be expected, and the ability of the algorithm to deal with increasing search space sizes improves only slightly. SHC demonstrates a large increase in performance and a greater ability to cope with more optima (a reduced standard deviation) post-tuning. The parameter configurations selected for the number of local optima, the ratio of local optima *and* the smoothness all have a neighbourhood size of around 50% of the search space size. We suggest that the performance improvement for all of these characteristics is actually derived from the algorithm having configured itself properly for the search space size used as a default for all other characteristics, rather than tuning itself to best perform on any specific characteristic.

## Conclusions and Future Work

In this paper we have built on previous studies of the performance of nature-inspired algorithms on fitness landscapes with different characteristics. Earlier work explored ‘out of the box’ parameter configurations, and we further develop this by using an automated parameter configuration methodology. This allows us to study the effect of tuning on different algorithms, contributing significantly to the debate on

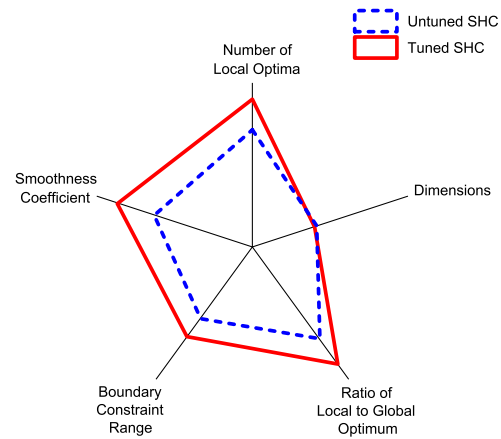


Figure 7: Summary of results for **Stochastic Hill-Climbing Algorithm**.

when and how it is beneficial to tune specific algorithms.

We observe that algorithms broadly fall into three categories: algorithms which *do not/sometimes/always* benefit from tuning by F-Racing. Dimensionality often offers the most significant improvement post-tuning in algorithms, particularly those with parameters that increase the *breadth* of search space (swarming algorithms are significantly better here than evolutionary algorithms). The methodology presented here is easy to implement, is computationally inexpensive, and offers considerably more information on the performance of an algorithm than using a standard set of benchmark problems. We hope that it will offer a framework for the experimental comparison of nature-inspired algorithms, as well as a useful set of heuristics for practitioners to use in order to decide when and how to tune their methods. Future work will focus on a comparative study of tuning techniques (i.e., in addition to F-Racing), and the application of our insights to the predictive performance ranking of methods on given problems.

## References

- Adenso-Diaz, B. and Laguna, M. (2006). Fine-tuning of algorithms using fractional experimental designs and local search. *Operations Research*, 54(1):99–114.
- Akay, B. and Karaboga, D. (2009). Parameter tuning for the artificial bee colony algorithm. In Nguyen, N., Kowalczyk, R., and Chen, S.-M., editors, *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, volume 5796 of *Lecture Notes in Computer Science*, pages 608–619. Springer Berlin Heidelberg.
- Bäck, T. and Schwefel, H.-P. (1993). An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, 1(1):1–23.
- Barr, R., Golden, B., and Kelly, J. (1995). Designing and reporting

- on computational experiments with heuristic methods. *Journal of Heuristics*, 1:9–32.
- Beyer, H.-g. and Schwefel, H.-p. (2002). Evolution strategies. *Natural Computing*, 1:3–52.
- Birattari, M. (2009). *Tuning metaheuristics: a machine learning perspective*. Springer.
- Birattari, M., Stützle, T., Paquete, L., and Varrentrapp, K. (2002). A racing algorithm for configuring metaheuristics. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '02*, pages 11–18, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Birattari, M., Yuan, Z., Balaprakash, P., and Stützle, T. (2010). F-race and iterated f-race: An overview. *Experimental methods for the analysis of optimization algorithms*, pages 311–336.
- Brabazon, A. and O'Neill, M. (2006). *Biologically inspired algorithms for financial modelling*. Springer-Verlag.
- Brownlee, J. (2011). *Clever Algorithms: Nature-Inspired Programming Recipes*. Lulu.
- Cant-Paz, E. (2001). Migration policies, selection pressure, and parallel evolutionary algorithms. *Journal of Heuristics*, 7(4):311–334.
- Crossley, M., Nisbet, A., and Amos, M. (2013). Fitness landscape-based characterisation of nature-inspired algorithms. In Tomassini, M., Antonioni, A., Daolio, F., and Buesser, P., editors, *Proceedings of the 11th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA'13)*, Lausanne, Switzerland, April 4-6, 2013. *Lecture Notes in Computer Science*, Vol. 7824, pages 110–119. Springer.
- Eiben, A. E. and Smit, S. K. (2011). Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, 1(1):19–31.
- Gallagher, M. and Yuan, B. (2006). A general-purpose tunable landscape generator. *IEEE Transactions on Evolutionary Computation*, 10(5):590–603.
- Geem, Z. and Kim, J. (2001). A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2):60–68.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.
- Hansen, N. and Kern, S. (2004). Evaluating the cma evolution strategy on multimodal test functions. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 282–291. Springer.
- Hendtlass, T. (2009). Particle swarm optimisation and high dimensional problem spaces. In *2009 IEEE Congress on Evolutionary Computation, CEC'09*, pages 1988–1994. IEEE.
- Horn, J. and Goldberg, D. (1994). Genetic algorithm difficulty and the modality of fitness landscapes. In *Foundations of Genetic Algorithms 3*.
- Jones, T. and Forrest, S. (1995). Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 184–192.
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings. ...*, pages 1942–1948.
- Koster, C. and Beney, J. (2007). On the importance of parameter tuning in text categorization. *Perspectives of Systems Informatics*, pages 270–283.
- Kukkonen, S. and Lampinen, J. (2005). GDE3: The third Evolution Step of Generalized Differential Evolution. *2005 IEEE Congress on Evolutionary Computation*, pages 443–450.
- Leung, F. H., Lam, H., Ling, S., and Tam, P. K. (2003). Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *Neural Networks, IEEE Transactions on*, 14(1):79–88.
- Lobo, F. G., Lima, C. F., and Michalewicz, Z. (2007). *Parameter setting in evolutionary algorithms*. Springer Verlag.
- Malan, K. M. and Engelbrecht, A. P. (2009). Quantifying ruggedness of continuous landscapes using entropy. In *2009 IEEE Congress on Evolutionary Computation*, pages 1440–1447. IEEE.
- Maron, O. and Moore, A. (1993). Hoeffding races: Accelerating model selection search for classification and function approximation. *Robotics Institute*, page 263.
- Maron, O. and Moore, A. W. (1997). The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11:193–225.
- Merz, P. (2000). Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *Evolutionary Computation, IEEE*, 4(4):337–352.
- Morgan, R. and Gallagher, M. (2010). When does dependency modelling help? Using a randomized landscape generator to compare algorithms in terms of problem structure. In et al Schaefer, R., editor, *PPSN XI*, pages 94–103. Springer-Verlag.
- Nannen, V., Smit, S. K., and Eiben, A. E. (2008). Costs and benefits of tuning parameters of evolutionary algorithms. In *Parallel Problem Solving from Nature-PPSN X*, pages 528–538. Springer.
- Passino, K. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3):52–67.
- Pham, D., Ghanbarzadeh, A., and Koc, E. (2006). The Bees Algorithm: A Novel Tool for Complex Optimisation Problems. In Pham, D., Eldukhri, E., and Soroka, A., editors, *Intelligent Production Machines and Systems*, pages 454–459.
- Ridge, E. and Kudenko, D. (2010). Tuning an algorithm using design of experiments. In *Experimental methods for the analysis of optimization algorithms*, pages 265–286. Springer.
- Smit, S. K. and Eiben, A. E. (2009). Comparing parameter tuning methods for evolutionary algorithms. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 399–406. IEEE.
- Yuan, B. and Gallagher, M. (2004). Statistical racing techniques for improved empirical evaluation of evolutionary algorithms. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 172–181. Springer.

# A hybrid genetic/immune strategy to tackle the multiobjective quadratic assignment problem

Arnaud ZINFLOU<sup>1</sup>, Caroline GAGNÉ<sup>2</sup> and Marc GRAVEL<sup>2</sup>

<sup>1</sup> Measurement and information systems division  
Institut de recherche d'Hydro-Québec – IREQ  
1800, Lionel-Boulet, Varennes, Canada, J3X 1S1  
zinflou.arnaud@ireq.ca

<sup>2</sup> Université du Québec à Chicoutimi - UQAC  
555 boulevard de l'Université, Chicoutimi, Canada G7H 2B1  
{caroline\_gagne,mgravel}@uqac.ca

## Abstract

The Genetic Immune Strategy for Multiple Objective Optimization (GISMOO) is a hybrid algorithm for solving multiobjective problems. The performance of this approach has been assessed using a classical combinatorial multiobjective optimization benchmark: the multiobjective 0/1 knapsack problem (MOKP) [1] and two-dimensional unconstrained multiobjective problems (ZDT) [2]. This paper shows that the GISMOO algorithm can also efficiently solve the multiobjective quadratic assignment problem (mQAP). A performance comparison carried out using well-known published algorithms and shows GISMOO to advantage.

## Introduction

Even today, most of the work on optimization treats problems with a single objective to optimize [3, 4]. However, practical contexts have a multiobjective nature inherent in various performance measures. For this type of problems, there is generally no ideal solution which gives optimality for all the objectives. This is why the optimal solution concept becomes less relevant and is replaced by Pareto-optimality, where we obtain a set of solutions giving us a compromise among the different objectives. The solutions cannot be prioritized except by the decision maker's preferences. Multiobjective problems also often have a very large feasible solution set and are characterized by repetitive decisions. Consequently, we can define two goals in multiobjective optimization: (i) to discover solutions as close to the Pareto-optimal solutions as possible; and (ii) to find solutions as diverse as possible in the solution set thereby obtained. Satisfying these two goals is a challenge for any multiobjective algorithm [5-10].

Many authors have concluded that a promising research model for multiobjective problem solving is that of metaheuristics adaptation [11, 12]. In fact, these algorithms are among the greatest achievements of modern operations research, especially in solving large and complex real problems [13-15]. According to Whitacre [13], the growing interest in metaheuristics in the last few years is due to the successful adaptation of these algorithms to specific problems and hybrid approaches.

Many of the algorithms proposed for multi-objective problems are Evolutionary Algorithms (EA) [6, 8, 16-19]. This is so, doubtlessly because EA's can traverse a large search space to generate an approximation of the Pareto-optimal front in a single optimization step [20]. One of these algorithms, proposed by [21], is a hybrid between a Genetic Algorithm (GA) and an Artificial Immune System (AIS). This approach, called GISMOO, has a small number of parameters to calibrate, is easy to implement, and has been shown to be efficient in solving classical benchmarks in both discrete and continuous optimization.

The goal of this paper is to deepen the understanding of the Quadratic Assignment Problem (QAP) from the Pareto viewpoint, by adapting the GISMOO algorithm to solve this problem. On the one hand, we mean to show that this is an interesting approach in the solution of the multiobjective Quadratic Assignment Problem (mQAP). On the other hand, because only a few workers have treated this problem from a Pareto viewpoint, we wish to compare the performance of GISMOO with that of other known algorithms.

The remainder of this paper is organized as follows. Section II briefly describes the mQAP. In Section III we present the GISMOO algorithm in order to solve the problem in a Pareto sense. Section IV details the numerical experiments carried out in this paper. Section V compares the experimental results of GISMOO with those of two other evolutionary algorithms well known in the literature: NSGAII and PMS<sup>MO</sup>. The last section offers some concluding remarks.

## The multiobjective Quadratic Assignment Problem (mQAP)

The Quadratic Assignment Problem (QAP), defined by Koopmans and Beckmann [22] in 1957, is one of the most well known and widely studied problems of combinatorial optimization. It consists of assigning  $n$  interconnected facilities (factories, warehouses, etc.) to  $n$  locations in such a way as to minimize the sum of the product flows over the distances. The problem can be formulated as follows:

$$\text{Minimise } C(\pi) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{\pi_i \pi_j} \quad (1)$$

where  $n$  is the number of locations/facilities,  $a_{ij}$  is the distance between location  $i$  and location  $j$ ,  $b_{ij}$  is the flow between facilities  $i$  and  $j$ , and  $\pi_i$  is the location of facility  $i$  in the permutation  $\pi \in \Omega$  where  $\Omega$  is the problem's solution space. The QAP belongs to the class of NP-hard problems [23, 24].

The multiobjective Quadratic Assignment Problem (mQAP) was formalized in 2002 [25]. This particular extension of the QAP considers several flow matrices between any two facilities. The mQAP can therefore model the situation of installations where the management of several types of products is a concern. For example, the flow of doctors, patients, visitors, pharmaceutical products, equipment, etc., between different facilities can be considered in the implantation of a (new) hospital. The mQAP can be formalized by the following equations:

$$\text{Minimise } \{C(\pi)\} = \{C^1(\pi), \dots, C^m(\pi)\}, \pi \in \Omega \quad (2)$$

with

$$C^k(\pi) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{\pi_i \pi_j}^k, k = 1, \dots, m \quad (3)$$

where  $m$  is the number of objectives (i.e. flow types),  $C(\pi)$  is the objective function vector to be simultaneously optimized,  $b_{\pi_i \pi_j}^k$  is the  $k$ -th flow between the facilities  $\pi_i$  et  $\pi_j$ , and *minimizing* means finding all non-dominated points [23]. Some research focuses on the resolution of mQAP [26-31].

## Genetic Immune Strategy for MultiObjective Optimization (GISMOO)

GISMOO is a hybrid Pareto GA/AIS algorithm which offers an original iterative process in two phases: a Genetic phase and an Immune phase. New solutions (also called descendants) are then obtained from the offspring creation using classical genetic operators and from the creation of clones according to the AIS cloning selection principle.

As for most multiple objective evolutionary algorithms, one of the main difficulties in solving multiple objective problems is the performance assignment. Indeed, the quality of a solution in multiobjective optimization depends on the evaluation of multiple incommensurable and often competing objective functions. Then, instead of dealing with the real objective functions the performance assignment of a Pareto EA can schematically be viewed as being composed of two factors: a *dominance factor* and an *isolation factor* [9]. The first factor evaluates the dominance level of a given solution in Pareto sense and the second factor measures the density of solutions surrounding a given solution. Even if the performance assignment of most Pareto EAs is done using these two factors, each algorithm calculates the two factors in different ways.

Regarding GISMOO, the dominance factor is calculated in two steps as proposed in [9]. The first step consists of assigning to each individual  $x$  of the Parent population ( $POP$ ) combined with the Descendant population ( $Q$ ), a strength  $S(x)$  corresponding to the number of solutions dominated by  $x$ . A solution  $x$  is said to *dominate* a solution  $y$  if and only if  $\exists i \in Z \mid f_i(x) < f_i(y)$ , and  $\forall j \in Z, j \neq i \mid f_j(x) \leq f_j(y)$  where  $f_i(\cdot)$  indicates the value of the solution for the objective  $i$  and  $Z$  indicates the number of objectives to minimize. Using the value of  $S(x)$ , the dominance factor of an individual  $x$ , called  $R^+(x)$ , is determined in GISMOO using Equation (4) below, where  $\succ$  indicates a dominance relationship of  $y$  with respect to  $x$ .

$$R^+(x) = \begin{cases} \frac{S(x)}{1 + 2 * S(x)} & \text{if } \sum_{y \in POP \cup Q, y \succ x} S(y) = 0 \\ \sum_{y \in POP \cup Q, y \succ x} S(y) & \end{cases} \quad (4)$$

In this way, for the non-dominated individuals  $x$  for which  $\sum_{y \in POP \cup Q, y \succ x} S(y) = 0$ , the dominance factor value is not

equal to 0, but rather between 0 and 0.5 according to the number of solutions it dominates. For the non-dominated solutions, this computation of the dominance factor allows us to better take into account the distribution of dominated solutions of  $POP$  and  $Q$  within the search space. Thus, the computation of the dominance factor favours non-dominated solutions situated in the less well explored regions.

The isolation factor, for its part, is based on the space metric  $sp$  introduced by Schott [32], which measures the distance  $Dist(x)$  between a given individual  $x$  and its closest neighbour  $y$  (with  $x \neq y$ ) as indicated in Equation (5).

$$Dist(x) = \min_{y \in POP \cup Q} \left[ \sqrt{(f_1(x) - f_1(y))^2 + \dots + (f_z(x) - f_z(y))^2} \right] \quad (5)$$

It should be noted that, in GISMOO, the isolation factor value is not directly added to the dominance factor value. In fact, the fitness of an individual in GISMOO is obtained using the dominance factor. and in case of a tie between individuals, the tie is broken using the isolation factor.

GISMOO's outline is shown in Algorithm 1. The algorithm starts building an initial population ( $POP_0$ ) of size  $N_1$  randomly or using greedy heuristics.

The main loop of GISMOO (lines 3-21) begins with the Genetic phase (lines 4-8) and generates  $N/2$  offspring. This phase consists of the classical operations of a GA: selection, crossover and mutation. Notice that the selection procedure used in GISMOO is a binary tournament selection. In addition, even if two offspring are created during the recombination, only the best of the two is added to the Descendant population  $Q$ . It is important to mention that no crossover probability is needed in the Genetic phase of GISMOO, because the number of offspring to generate is related to the Parent population size. However, a mutation probability ( $p_m$ ) is used to determine whether the generated offspring will be mutated or not (line 6). The crossover operator used in the Genetic phase to solve the mQAP problem is a cycle crossover operator [33] as proposed by [27]



for the same problem. For the mutation operator we choose to use a classical inversion operator.

---

**Algorithm 1** : Outline of GISMOO procedure
 

---

```

1: Create an initial Parent population  $POP_0$  of size  $N$ 
2: Initialize  $t$  to 1
3: While no stopping rules is invoked
4:   While  $|Q_t| < N/2$ 
5:     Select and recombine  $P_1$  and  $P_2 \in POP_t$  to obtain  $E_1$  and  $E_2$ 
6:     Mutate  $E_1$  and  $E_2$  according to mutation probability  $p_m$ 
7:     Evaluate  $E_1$  and  $E_2$  and add the best offspring found in  $Q_t$ 
8:   End While Genetic Phase
9:   Rank non-dominated solutions of  $POP_t$ 
10:  For each non-dominated solution  $x \in POP_t$  do
11:    Calculate  $nb\_clones_x$  to produce for  $x$  according to Eq. (6)
12:     $cpt = 0$ 
13:    While  $cpt < nb\_clones_x$ 
14:      Create 2 clones  $C_1^{clo}$  et  $C_2^{clo}$  using  $x$ 
15:      Create  $C_1^{hyp\alpha}$  (and  $C_2^{hyp\beta}$ ) by hyper-mutation  $\alpha(\beta)$  on  $C_1(C_2)$ 
16:      Evaluate  $C_1^{hyp\alpha}$  and  $C_2^{hyp\beta}$  and add the best of the two in  $Q_t$ 
17:       $cpt = cpt + 1$ 
18:    End While
19:  End For Immune Phase
20:  Copy the  $N$  first solutions of  $POP_t \cup Q_t$  into  $POP_{t+1}$ 
21:   $t = t + 1$ 
22: End While
    
```

---

Thereafter, the Immune phase (lines 9-19) adds  $N/2$  solutions to the Descendant population  $Q$ . These solutions are generated using the cloning selection principle introduced by De Castro and Timmis [34]. This principle is used to model the fact that only the best “antibodies” will proliferate in the population. In GISMOO, antibodies correspond to the non-dominated solutions of the current population  $POP$ . The number of clones to create for each non-dominated solution  $x$  ( $nb\_clones_x$ ) is related to the isolation factor as shown in Equation (6) :

$$nb\_clones_x = \text{round} \left[ \left( \text{Dist}(x) * \frac{N}{2} \right) / \sum_{x=1}^{nbSolND} \text{Dist}(x) \right] \quad (6)$$

where  $nbSolND$  indicates the total number of non-dominated solutions in the current population and  $\text{Dist}(x)$  corresponds to the isolation factor of an individual  $x$  as shown previously. The *round* function returns a number rounded to the nearest integer.

We note that a non-dominated solution in a less crowded region of the search space will generate a greater number of clones. This calculation is then dynamically adjusted during the iterative process of the algorithm and does not require the setting of additional parameters.

Once the number of clones to generate for an individual  $x$  is calculated, we generate two copies ( $C_1^{clo}, C_2^{clo}$ ) of  $x$ . Thereafter, the two clones are hyper-mutated using two different mutation operators  $\alpha$  and  $\beta$ , depending on the problem to solve, in order to obtain two mutated clones ( $C_1^{hyp\alpha}, C_2^{hyp\beta}$ ). The hyper-mutation rate to apply to all clones

generated for an individual  $x$  is set according to their rank. Indeed, for the first ten individuals with the highest rank the hyper-mutation rate is set at  $Z$ , the number of objectives to minimize. For each ten individuals the hyper-mutation rate is increased by one until we reach all non-dominated solutions. The two mutation operators used to solve the mQAP problem are respectively a classical inversion and a simple swap. We then compare the two obtained solutions, only adding the best in the Descendant population  $Q$ . This process is repeated until the total number of clones to be generated is reached.

After the Genetic and Immune phases, an elitist replacement of the population in order to keep the  $N$  best solutions of the combined Parent and Descendant population (line 20) is made. The replacement strategy used is a  $(\lambda + \mu)$  type of deterministic replacement where  $\lambda$  indicates the size of the Parent population and  $\mu$  the size of the Descendant population. In the proposed approach, we have  $\lambda = \mu = N$ . Finally, notice that GISMOO also used an archive in order to store non-dominated solutions found during the search. For a detailed description of GISMOO, the reader can consult Zinflou *et al.* [21].

## Numerical experiments

The focus of the experiments was on comparing the performance of GISMOO to those of two state-of-the-art algorithms: NSGAI [6] and PMS<sup>MO</sup> [9]. For a detailed description of these two evolutionary algorithms, the reader can respectively consult [6] [9].

### Test problems

We used a set of 22 benchmark mQAP instances to test the performance of the three evolutionary multiple objective algorithms. These test instances were generated by Knowles and Corne [25] and are available at <http://dbkgroup.org/knowles/mQAP/>. The number of locations, number of objectives and the category of each instance are indicated in Table 1.

### Experimental conditions

All the algorithms used in this work were implemented in C++ and compiled with Visual Studio .Net 2010, using the same main data structures in our implementation. All the algorithms use the same population size set at 100 individuals, and the same computational time for the same test instance. The computational times used for the instances with 10, 20, and 30 locations are set to 10, 20, and 30 seconds, respectively [27].

The parameter settings for NSGAI, PMS<sup>MO</sup> and GISMOO are determined empirically according to our previous works [1, 21] and the work of [27]. For the three algorithms, the mutation probability ( $p_m$ ) was set at 0.06. It is important to mention here that no crossover probability is required in the GISMOO algorithm. For the two other algorithms, the crossover probability was set at 1. The population size  $N$  is the same for all algorithms and was set at 100. To perform a recombination, all the algorithms in this paper use a cycle crossover operator [33]. As a mutation operator, each algorithm uses a classical inversion.

| Problem        | Instance category | # of locations | # of objectives |
|----------------|-------------------|----------------|-----------------|
| KC10-2fl-1uni* | Uniform           | 10             | 2               |
| KC10-2fl-2uni* | Uniform           | 10             | 2               |
| KC10-2fl-3uni* | Uniform           | 10             | 2               |
| KC20-2fl-1uni  | Uniform           | 20             | 2               |
| KC20-2fl-2uni  | Uniform           | 20             | 2               |
| KC20-2fl-3uni  | Uniform           | 20             | 2               |
| KC30-3fl-1uni  | Uniform           | 30             | 3               |
| KC30-3fl-2uni  | Uniform           | 30             | 3               |
| KC30-3fl-3uni  | Uniform           | 30             | 3               |
| KC10-2fl-1rl * | Real-like         | 10             | 2               |
| KC10-2fl-2rl * | Real-like         | 10             | 2               |
| KC10-2fl-3rl * | Real-like         | 10             | 2               |
| KC10-2fl-4rl * | Real-like         | 10             | 2               |
| KC10-2fl-5rl*  | Real-like         | 10             | 2               |
| KC20-2fl-1rl   | Real-like         | 20             | 2               |
| KC20-2fl-2rl   | Real-like         | 20             | 2               |
| KC20-2fl-3rl   | Real-like         | 20             | 2               |
| KC20-2fl-4rl   | Real-like         | 20             | 2               |
| KC20-2fl-5rl   | Real-like         | 20             | 2               |
| KC30-3fl-1rl   | Real-like         | 30             | 3               |
| KC30-3fl-2rl   | Real-like         | 30             | 3               |
| KC30-3fl-3rl   | Real-like         | 30             | 3               |

Table 1: Characteristics of the test suites used

The computational experiments were run on a HP Z600 workstation with 2.13 GHz quad core Intel Xeon processor and 4 Gb of RAM, with Windows XP. Each instance was solved 30 times for each algorithm with random seeds.

### Performance assessment

The test procedure for performance assessment uses the generational distance (GD) metric [27]. This metric evaluates the average distance between the obtained non-dominated set ( $E$ ) and the reference set  $P^*$  of Pareto-optimal solutions or a very good approximation. The GD metric is computed using the following equation:

$$GD(E, P^*) = \frac{1}{|E|} \sum_{u \in E} \min\{dist(u, v) \mid v \in P^*\} \quad (7)$$

where  $dist(u, v)$  is the Euclidean distance (in objective spaces) between the solution  $u \in E$  and the nearest member  $v$  in  $P^*$  and  $|E|$  represents the cardinality of the set  $E$ . The smaller the value of this metric, the better the convergence toward the Pareto-optimal front. When all obtained solutions lie exactly on  $P^*$  chosen solutions, this metric has a value of zero.

The GD allows us to compare the preliminary results obtained by GISMOO to those of well known algorithms. However, it is obvious that additional performance assessments are needed in multiobjective optimization in order to prove conclusively the superiority of an algorithm. In this paper, the reference sets for the 10 location instances are the true Pareto front available at <http://dbkgroup.org/knownles/mQAP/>. From the approximation sets found by the three algorithms for each 20 and 30 location instances, the set containing only non-dominated solutions was computed and used as the reference set.

In addition to GD, performance assessment was also undertaken using PISA and the guidelines from [35] and [36]. Consider for example the comparison between GISMOO and PMS<sup>MO</sup> on a problem. First, the bounds of approximation sets of both algorithms were calculated so that the approximation sets could be normalized to the interval [1, 2]. After that, a dominance rank was calculated for each of the 60 approximation sets by simply counting the number of approximation sets that are better than the observed one. The Mann-Whitney rank sum test was used to discover if there are significant differences between the dominance ranks of the two algorithms.

## Results and discussion

Table 2 presents the average GD values obtained by each algorithm for the 22 mQAP instances. The first column of each table indicates the name of the problem. In this column a “\*” right after the name of the problem denotes the problems on which the GD was calculated using true Pareto optimal solutions, while “†” indicates that we used an approximation set to calculate the GD values. The following columns respectively show the average results over 30 runs for each algorithm. With regard to the average best results obtained, they are indicated by a shaded area.

Table 2 shows that GISMOO globally outperforms the other algorithms. Indeed, GISMOO’s GD metric is better than that of the other two algorithms in every one of the 22 tested instances.

| Problem        | Algorithm         |         |         |
|----------------|-------------------|---------|---------|
|                | PMS <sup>MO</sup> | NSGAII  | GISMOO  |
| KC10-2fl-1uni* | 1703.65           | 2248.78 | 0       |
| KC10-2fl-2uni* | 4671.3            | 10032.7 | 0       |
| KC10-2fl-3uni* | 240.129           | 321.019 | 56.4565 |
| KC20-2fl-1uni† | 9064.52           | 15357.5 | 2329.48 |
| KC20-2fl-2uni† | 19008.3           | 30414.5 | 8999.31 |
| KC20-2fl-3uni† | 1751.35           | 3095.25 | 551.016 |
| KC30-3fl-1uni† | 6212.65           | 12861.4 | 2405.68 |
| KC30-3fl-2uni† | 23244.3           | 38251.6 | 5880.51 |
| KC30-3fl-3uni† | 3400.69           | 5398.18 | 890.908 |
| KC10-2fl-1rl * | 36076.6           | 83310.2 | 4534.06 |
| KC10-2fl-2rl * | 85563.9           | 104218  | 0       |
| KC10-2fl-3rl * | 18781.2           | 84171.2 | 2014.34 |
| KC10-2fl-4rl * | 5578.69           | 16619.8 | 0       |
| KC10-2fl-5rl*  | 58147.3           | 99401.8 | 2797.36 |
| KC20-2fl-1rl † | 415416            | 694862  | 64192.5 |
| KC20-2fl-2rl † | 206671            | 345675  | 29807.8 |
| KC20-2fl-3rl   | 162788            | 245988  | 16266.3 |
| KC20-2fl-4rl†  | 371847            | 706505  | 84916.1 |
| KC20-2fl-5rl†  | 1373590           | 2529950 | 222566  |
| KC30-3fl-1rl†  | 172467            | 314176  | 51039.8 |
| KC30-3fl-2rl†  | 242664            | 522926  | 56675.9 |
| KC30-3fl-3rl†  | 195027            | 250777  | 23985.2 |

 Table 2: Average GD values of non-dominated solutions found by PMS<sup>MO</sup>, NSGAII and GISMOO in 30 runs

In Table 3 and 4, we summarize the outcome of the Mann-Whitney rank sum tests for a significance level  $\alpha$  of 0.0125. In these tables, the “ $\uparrow$  p-value” ( $\downarrow$  p-value) denotes the problems, on which GISMOO is significantly better (worse) than the other algorithm in comparison, while “ $\leftrightarrow$ ” indicates there are no significant differences between the two algorithms.

| Problem        |            |
|----------------|------------|
| KC10-2fl-1uni* | $\uparrow$ |
| KC10-2fl-2uni* | $\uparrow$ |
| KC10-2fl-3uni* | $\uparrow$ |
| KC20-2fl-1uni† | $\uparrow$ |
| KC20-2fl-2uni† | $\uparrow$ |
| KC20-2fl-3uni† | $\uparrow$ |
| KC30-3fl-1uni† | $\uparrow$ |
| KC30-3fl-2uni† | $\uparrow$ |
| KC30-3fl-3uni† | $\uparrow$ |
| KC10-2fl-1rl * | $\uparrow$ |
| KC10-2fl-2rl * | $\uparrow$ |
| KC10-2fl-3rl * | $\uparrow$ |
| KC10-2fl-4rl * | $\uparrow$ |
| KC10-2fl-5rl*  | $\uparrow$ |
| KC20-2fl-1rl † | $\uparrow$ |
| KC20-2fl-2rl † | $\uparrow$ |
| KC20-2fl-3rl   | $\uparrow$ |
| KC20-2fl-4rl†  | $\uparrow$ |
| KC20-2fl-5rl†  | $\uparrow$ |
| KC30-3fl-1rl†  | $\uparrow$ |
| KC30-3fl-2rl†  | $\uparrow$ |
| KC30-3fl-3rl†  | $\uparrow$ |

Table 3: Outcomes of the Mann-Whitney rank sum test on dominance ranking for GISMOO and NSGAI

| Problem        |            |
|----------------|------------|
| KC10-2fl-1uni* | $\uparrow$ |
| KC10-2fl-2uni* | $\uparrow$ |
| KC10-2fl-3uni* | $\uparrow$ |
| KC20-2fl-1uni† | $\uparrow$ |
| KC20-2fl-2uni† | $\uparrow$ |
| KC20-2fl-3uni† | $\uparrow$ |
| KC30-3fl-1uni† | $\uparrow$ |
| KC30-3fl-2uni† | $\uparrow$ |
| KC30-3fl-3uni† | $\uparrow$ |
| KC10-2fl-1rl * | $\uparrow$ |
| KC10-2fl-2rl * | $\uparrow$ |
| KC10-2fl-3rl * | $\uparrow$ |
| KC10-2fl-4rl * | $\uparrow$ |
| KC10-2fl-5rl*  | $\uparrow$ |
| KC20-2fl-1rl † | $\uparrow$ |
| KC20-2fl-2rl † | $\uparrow$ |
| KC20-2fl-3rl   | $\uparrow$ |
| KC20-2fl-4rl†  | $\uparrow$ |
| KC20-2fl-5rl†  | $\uparrow$ |
| KC30-3fl-1rl†  | $\uparrow$ |
| KC30-3fl-2rl†  | $\uparrow$ |
| KC30-3fl-3rl†  | $\uparrow$ |

Table 4: Outcomes of the Mann-Whitney rank sum test on dominance ranking for GISMOO and PMS<sup>MO</sup>

The Mann-Whitney rank sum test results also show that GISMOO globally outperforms the other two genetic algorithms. Indeed, GISMOO is never worse than NSGAI or PMS<sup>MO</sup> in any of the 22 mQAP instances. In fact, GISMOO is significantly better than NSGAI and PMS<sup>MO</sup> on all the instances. These results confirm the results obtained with the GD metric.

Beside the GD and Mann-Whitney rank sum test results, Fig. 1 and 2 graphically represent the solution sets found by GISMOO, NSGAI and PMS<sup>MO</sup> on a typical run using respectively two ten location problems: KC10-2fl-1uni and KC10-2fl-1rl. The KC10-2fl-1uni is a uniform instance while KC10-2fl-1rl is a real-like instance. In these two particular examples, we notice that GISMOO globally outperforms NSGAI and PMS<sup>MO</sup>. Indeed, in both cases the curves proposed by GISMOO dominated and are more extended than those of NSGAI and PMS<sup>MO</sup>. We also notice that for these two problems the approximation sets found by GISMOO are very close to the true Pareto optimal solution. In fact, for the problem KC10-2fl-1uni all the solutions found by GISMOO are also in the Pareto. These graphs confirm the results of the GD metric and the ability of our approach to find diversified solutions and to efficiently explore the solution space.

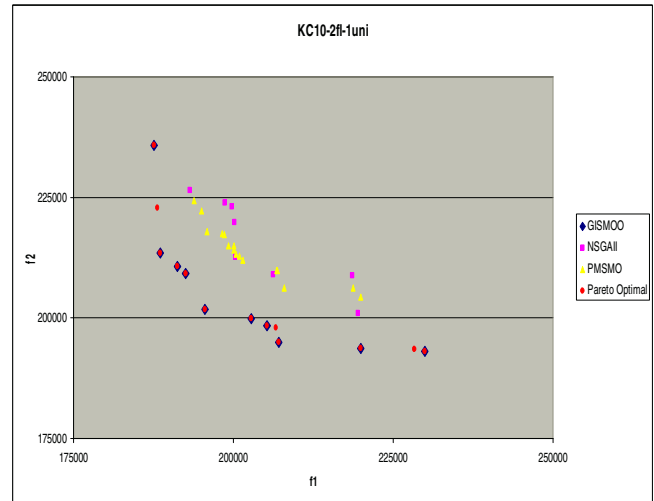


Figure 1: Non-dominated sets of a typical run of GISMOO, NSGAI and PMS<sup>MO</sup>

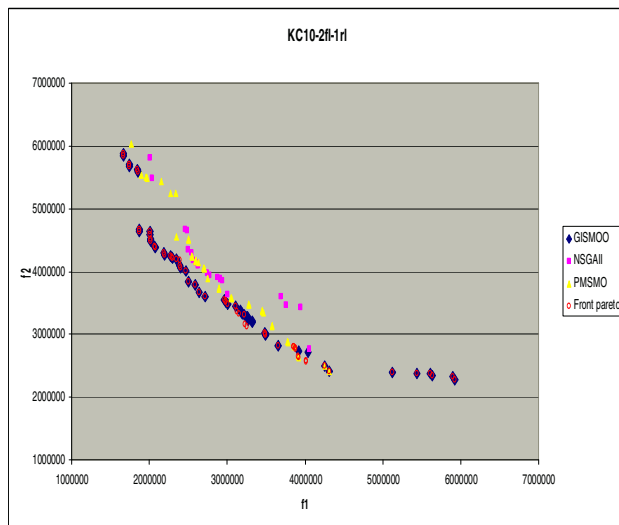


Figure 2: Non-dominated sets of a typical run of GISMOO, NSGAII and PMSMO on the KC10-2fl-1rl problem

## Conclusion

In this paper, we compared our GISMOO algorithm with two well known multiobjective algorithms (NSGAII, PMSMO) on twenty-two state-of-the-art benchmark mQAP problems. The biggest difference between GISMOO and other two algorithms lies in the environmental selection and the way in which the immune metaphor is used in a Pareto GA to identify and emphasize the solutions located in less crowded regions found during the iteration process of the algorithm. The preliminary experimental results obtained on the benchmark problems considered here have shown that our approach is efficient: on every problem, GISMOO outperformed the other algorithms with regard to the quality indicators used here. These results confirm those obtained in [21] and [2] respectively for other combinatorial and continuous optimization problems.

On the basis of these results we suggest that GISMOO is an efficient tool to solve the mQAP problem. In future work, we will seek to extend the numerical experiments to other performance metrics. We will also seek to extend the application field of GISMOO to other multiobjective problems in industry and elsewhere.

## References

- [1] A. Zinlou, C. Gagné, and M. Gravel, "Solving multiple-objective optimization problems using GISSMO algorithm," in *World Congress on Nature and Biologically Inspired Computing (NABIC 2009)*, Coimbatore, India, 2009, pages 239-244.
- [2] A. Zinlou, C. Gagné, and M. Gravel, "GISMOO on Continuous Multiple-Objective Problems: a Comparison Study," in *Proceedings of the 2010 International Conference on Genetic and Evolutionary Methods (GEM'10)*, Las Vegas, 2010, pages 196-202.

- [3] T. Loukil, J. Teghem, and D. Tuytens, "Solving multi-objective production scheduling problems using metaheuristics," *European Journal of Operational Research*, vol. 161, pages 42-61, 2005.
- [4] R. Ruiz and J. A. Vázquez-Rodríguez, "The hybrid flow shop scheduling problem," *European Journal of Operational Research*, vol. 205, pages 1-18, 2010.
- [5] A. C. Coello Coello and G. T. Pulido, "Multiobjective Optimization using a Micro-genetic Algorithm," in *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 274-282, San Francisco, California, 2001.
- [6] K. Deb, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multiobjective Optimization : NSGA II," in *Parallel problem Solving from Nature – PPSN VI*, M. Schoenauer et al. (Eds), Springer Lecture Notes in Computer Science, pages 849-858, 2000.
- [7] J. D. Knowles and D. W. Corne, "The Pareto Archived Evolution Strategy : A New Baseline Algorithm for Multiobjective Optimisation," in *Congress on Evolutionary Computation*, Washington, 1999, pages 98-105.
- [8] J. D. Knowles and D. W. Corne, "The Pareto-Envelope based Selection Algorithm for Multiobjective Optimization," in *In Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, pages 839-848, 2000.
- [9] A. Zinlou, C. Gagné, M. Gravel, and W. L. Price, "Pareto memetic algorithm for multiple objective optimization with an industrial application," *Journal of Heuristics*, vol. 14, pages 313-333, 2008.
- [10] E. Zitzler, "Evolutionary Algorithms for Multiobjective Optimization," in *EUROGEN 2001 - Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems.*, 2001.
- [11] J. Teghem and A. Jaskiewicz, "Multiple objective metaheuristics for combinatorial optimization: a tutorial," in *the 5th Metaheuristics International Conference Kyoto*, Japon, 2003.
- [12] M. Ehrgott and X. Gandibleux, "A survey and annotated bibliography of multiobjective combinatorial optimization," *OR Spektrum*, vol. 22, pages 425-460, 2000.
- [13] J. M. Whitacre, "Survival of the flexible: explaining the recent popularity of nature-inspired optimization within a rapidly evolving world," *Computing*, vol. 93, pages 135-146, 2011.
- [14] H. Zang, S. Zang, and K. Hapeshi, "A Review of Nature-Inspired Algorithms," *Journal of Bionic Engineering*, vol. 7, pages S232-S237, 2010.
- [15] U. Derigs and S. Voß, "Meta-Heuristics - Theory, Applications and Software " *Annals of Operations Research*, vol. 131, pages 17-20, 2004.
- [16] V. Cutello, G. Narzisi, and G. Nicosia, "A Class of Pareto Archived Evolution Strategy Algorithms using Immune inspired Operators for Ab-Initio Protein Structure Prediction," in *Third European Workshop on Evolutionary Computation and Bioinformatics, EvoWorkshops 2005, EvoBio 2005*, Lausanne, Switzerland, 2005, pages 3449.
- [17] V. Cutello, G. Narzisi, and G. Nicosia, "A Multi-Objective Evolutionary Approach to the Protein Structure Prediction Problem," *Journal of the Royal Society Interface*, vol. 3, pages 139-151, 2006.



- [18] J. D. Knowles and D. W. Corne, "M-PAES : A Memetic Algorithm for Multiobjective Optimization," in *In Proceedings of the 2000 Congress on Evolutionary Computation*, 2000, pages 325-332.
- [19] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2 : Improving the Strength Pareto Evolutionary Algorithm," Technical Report 103, Computer Engineering and Communication Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland.2001.
- [20] D. Francisci, "Algorithmes évolutionnaires et optimisation multi-objectifs en data mining," Laboratoire informatique, signaux et systèmes de Sophia Antipolis UMR 60702002.
- [21] A. Zinlou, C. Gagné, and M. Gravel, "GISMOO: A New Hybrid Genetic/Immune Strategy for Multiple Objective Optimization," *Computers & Operations Research*, vol. 39, pages 1951-1968, 2012.
- [22] T. C. Koopmans and M. J. Beckmann, "Assignment problems and the location of economic activities," *Econometrica*, vol. 25, pages 53-76, 1957.
- [23] E. Çela, *The Quadratic Assignment Problem - Theory and Algorithms*. Boston, MA: Kluwer Academic Publishers, 1998.
- [24] M. S. Garey and D. S. Johnson, *Computer and Intractability : A Guide to the Theory of NP-Completeness*. New York: W.H. Freeman and Co., 1979.
- [25] J. Knowles and D. Corne, "Instance generators and test suites for the multiobjective quadratic assignment problem," presented at the Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Faro, Portugal, 2003.
- [26] R. O. Day, M. P. Kleeman, and G. B. Lamont, "Solving the Multi-objective Quadratic Assignment Problem Using a fast messy Genetic Algorithm.," in *Congress on Evolutionary Computation (CEC'2003)*, Piscataway, New Jersey, 2003, pages 2277-2283.
- [27] H. Li and D. Landa-Silva, "En Elitist GRASP Metaheuristic for the Multi-objective Quadratic Assignment Problem," in *5th International Conference on Evolutionary Multi-Criterion Optimization*, Heidelberg, 2009, pages 481-494.
- [28] J. D. Knowles and D. W. Corne, "Towards landscape analyses to inform the design of hybrid local search for the multiobjective quadratic assignment problem," in *Soft Computing Systems - Design, Management and Applications (HIS 2002)*, 2002, pages 271-279.
- [29] M. P. Kleeman, R. O. Day, and G. B. Lamont, "Analysis of a Parallel MOEA Solving the Multi-objective Quadratic Assignment Problem," in *GECCO*, 2004, pages 402-403.
- [30] R. O. Day and G. B. Lamont, "Multiobjective Quadratic Assignment Problem Solved by an Explicit Building Block Search Algorithm - MOMGA-IIa," in *Evocop 2005*, 2005, pages 91-100.
- [31] L. Paquete and T. Stützle, "A study of stochastic local search algorithms for the biobjective QAP with correlated flow matrices," *European Journal of Operational Research*, vol. 169, pages 943-959 2006.
- [32] Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," Master's thesis, Department of Aeronautics and Astronautics Massachusetts Institute of Technology, 1995.
- [33] I. M. Oliver, D. J. Smith, and J. R. C. Holland, "A study of permutation crossover operators on the traveling salesman problem," in *the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, Hillsdale, NJ, USA, 1987, pages 224-230.
- [34] L. N. De Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. London, 2002.
- [35] J. D. Knowles, L. Thiele, and E. Zitzler, "A tutorial on the performance assessment of stochastic multiobjective optimizers.," TIK-Report No. 214, Computer Engineering and Networks Laboratory2006.
- [36] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. D. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, 2003.

# Using explicit averaging fitness for studying the behaviour of rats in a maze

Ariadne de A. Costa<sup>1</sup>, Patrícia A. Vargas<sup>2</sup> and Renato Tinós<sup>3</sup>

<sup>1</sup> Department of Physics, FFCLRP, University of São Paulo, Ribeirão Preto, SP, Brazil

<sup>2</sup> School of Mathematical and Computer Science, Heriot-Watt University, Edinburgh, Scotland, UK

<sup>3</sup> Department of Computing and Mathematics, FFCLRP, University of São Paulo, Ribeirão Preto, SP, Brazil  
ariadne.ad@pg.ffclrp.usp.br

## Abstract

In this paper we study the performance of different evolutionary strategies based on explicit averaging. On a previous study, (Costa et al., 2012) proposed a probabilistic fitness function for an agent model based on neural networks and genetic algorithms employed to investigate the behaviour of rats in an elevated plus-maze (EPM). Differently from other computational models, the virtual rat proposed in (Costa et al., 2012) is not built based on experimental data comparisons with real rats, but, instead, is based on a behavioural model exploring the conflict between fear and anxiety. Despite the good results of the proposed agent, the effects of the uncertain fitness functions in the evolutionary learning process were not studied in the previous study. In our experiments we found significant differences in the performance of the genetic algorithm when the fitness of the individuals is sampled different times thus enabling us to define the best strategy for the studied problem.

Genetic algorithm, Uncertainty, Explicit averaging fitness, Elevated plus-maze, Rat

## Introduction

Uncertainty has been widely studied in different areas, like politics (Cioffi-Revilla, 1998; Carmignani, 2003), economics (Naceur Jabnoun and Yusuf, 2003; Pindyck, 2007; Baker et al., 2013), movies (Miller and Shamsie, 1999; Vany, 2004; Gil, 2008), sports (Jennet, 1984; Peel and Dennis, 1992; Knowles et al., 1992; Forrest and Simmons, 2002; Buraimo and Simmons, 2008), criminology (Lattimore and Witte, 1986; Goulas and Zervoyianni, 2013) and sociology (Marris, 1996). By reviewing the uncertainty literature, we find many concepts for this word. Early studies claim that uncertainty is product of unpredictability (Cyert and March, 1963) or environmental turbulence (Emery and Trist., 1965). In (Lawrence and Lorsch, 1967), uncertainty is described as the lack of knowledge in the decision making process and similarly, in (Duncan, 1972), uncertainty is the absence of information for decision making. Uncertainty is also faced as result of the complexity of influential variables (Galbraith, 1973).

Working with evolutionary computation, there are various kinds of uncertainties involved. According to (Jin and

Branke, 2005), these uncertainties may be classified into four groups:

1. the fitness function is subject to noise, which also may be comprehend as a problem of a partially observable environment;
2. the variables and/or the environmental parameters may change after simulation;
3. the fitness function is an approximation, which may cause errors;
4. the optimum of the problem is dynamic, so the optimizer has to seek the optimum continuously.

(Campi and Calafiore, 2004) adds another class to this list:

5. the optimization is based on sampling finite, small, number of instances (*scenarios*), normally because of the cost of obtaining each instance.

One of the classes of uncertainty problems that has been most studied is related to problems where the fitness function is subject to noise (Fitzpatrick and Grefenstette, 1988; Rattray and Shapiro, 1996; Sondahl and Stonedahl, 2010). Many of the solutions adopted to this class of problems investigate the influence of different selection schemes (Miller and Goldberg, 1996; Sano, 2002; Goschin et al., 2011) in the performance of the evolutionary algorithms. The problem studied here belongs to the first class of uncertain problems.

Uncertainty is also one of the greatest difficulties during decision making processes. The process of decision in rats navigating in mazes is widely investigated in animal behavioural studies (Salum et al., 2003; Walf and Frye, 2007). One of the mazes extensively used for this purpose is the elevated plus-maze (EPM). The EPM is an elevated plus-shaped maze composed of two opposed open arms and two opposed enclosed arms. The experiments using EPM are mostly derived from studies realized by Montgomery, in 1955, for the investigation of the conflict between fear and anxiety in rats during their exposition to a new environment (Montgomery, 1955). The conflict in the animal is

caused by sensations of curiosity and fear occurring simultaneously and understood as defence mechanisms (Graeff, 1990). Most of the studies of rats in the EPM are based on the results of Montgomery (Montgomery, 1955), which show that rats spend more time in the enclosed arms of the maze than in the open arms.

In the last years, some researches proposed computational models to simulate the rats behavior in the EPM (Salum et al., 2000; Giddings, 2002; Miranda et al., 2009; Shimo et al., 2010; Costa et al., 2012). In (Salum et al., 2000), artificial neural networks (ANNs), trained by competitive learning, control a virtual rat in a virtual EPM. The inputs of the ANN are: need for exploration, aversion repulsive stimuli, and spontaneous motor activity. In the paper, the authors also studied a completely open and a completely closed maze. (Miranda et al., 2009) proposes a very similar model. (Giddings, 2002) implements a computational model based on the fact that the rat generally remains in the direction that it already is. The probabilities of following the previous direction, or moving to other direction, are defined a priori for different parts of the maze.

Other approach is used in (Shimo et al., 2010; Costa et al., 2012), in which the virtual agent is obtained from a optimisation process. In this case, the agent is a mobile robot controlled by an ANN. The weights of the ANN are defined by a genetic algorithm (GA), which uses a fitness function based on the comparison of the trajectories of the individuals in a replica of the EPM and the trajectories of real rats in an EPM.

According to the authors' knowledge, (Costa et al., 2012) proposed the first model in which the virtual agent is obtained independently from experimental data obtained with real rats. In the same way of the model proposed in (Shimo et al., 2010; Costa et al., 2012), the computational agent (a virtual robot) is represented by an artificial neural network, whose weights are evolved by a genetic algorithm. The outputs of the ANN determine the next action of the virtual robot. The main contribution of (Costa et al., 2012) is a probabilistic fitness function that is not dependent of experimental data obtained with experiments using real rats, but is based on the relation of fear and anxiety related by Montgomery. The results of the experiments with real rats are employed only in the validation of the model. The experimental results showed that the virtual model was capable of reproducing the rats behaviour, when some parameters of the trajectories like number of entrances and time spent in each arm are considered. However, the effects of the uncertainty inherent to the fitness function were not studied on our previous study (Costa et al., 2012). The problem by using a probabilistic function, is that the comparison of candidate solutions can be unfair, causing bias in the selection of the individuals, and affecting the evolutionary process.

As mentioned in (Jin and Branke, 2005), one of the classes of uncertainty in evolutionary optimisation occurs

when the fitness function is noisy, and there are many types of strategies to deal with that uncertainty. One of them is explicit averaging, where the fitness of an individual is sampled a number of times. In this paper, we test different evolutionary strategies based on explicit averaging for the model proposed in (Costa et al., 2012), with the intention of reducing the effects of the uncertainty in the fitness function employed by the virtual robot. One problem with explicit averaging is the additional computational costs for the optimisation algorithm. This is particularly true for problems where the evaluation process is costly, like in robots. If the evaluation is costly, an option is to use a less accurate fitness function and another option is to calculate the average based on fewer independent individual evaluations (Sondahl and Stonedahl, 2010). However, this is not the case of our problem, as the fitness evaluation can be split in two parts: one to obtain the trajectory of the agent, which is costly, and another to compute the fitness based on this trajectory. Since the last one is not costly, we are able to make many independent evaluations to calculate the mean fitness for each individual without significantly changing the time required to evaluate an individual.

In next section we explain the computational methods for the virtual rat and the four strategies studied in this paper to deal with uncertainty. In Section III, we present experiments comparing the fitness and the time spent in the enclosed and open arms for the virtual rats obtained with these strategies. Conclusions and future directions are presented in Section IV.

## Methodology

In this paper, a computational agent (virtual robot) is employed to simulate the behaviour of a rat in the EPM. The virtual robot is controlled by a recurrent multilayer perceptron (Elman's network) with ten inputs (six sensors and four recurrent signals from the hidden layer), four neurons in the hidden layer, and four neurons in the output layer. The recurrence is important because it allows that previous inputs be stored in the internal neurons, acting like a memory. The sensors are placed around the robot in order to detect the walls of the EPM. The outputs of the ANN indicate the next robot's action (stay, turn left, turn right or go forward to the next position).

A genetic algorithm optimises the weights of the artificial neural network. This way, each individual of the GA corresponds to a chromosome composed by an array of integers, representing a possible solution in the fitness landscape. The initial population is randomly chosen.

The same virtual EPM proposed in (Salum et al., 2000) is employed in our work. Each arm of the plus-maze is divided in five positions plus the central position that links the four arms of the EPM, totalling 21 positions. The agent (rat or virtual robot) is evaluated by its navigation in this virtual EPM during a period of time corresponding to 5 minutes for

the real rat, or 300 time steps for the virtual rat.

The fitness function, proposed in (Costa et al., 2012) and employed here, is based on the conflict of fear and anxiety model (Montgomery, 1955). Two terms compose the fitness function: one for reward and other for punishment. The reward represents the curiosity of the rat in exploring not recently visited positions of the maze, while the punishment represents the exposure to damage. This last term is probabilistic, i.e., the same trajectory of the agent in the EPM can generate different values for this term. The fitness of individual  $\mathbf{x}$  is computed based on the trajectory of the agent controlled by the ANN with weights given by the chromosome of the individual in the virtual EPM. In this way, after selection and reproduction, each individual of the GA generates a trajectory in the virtual EPM and this trajectory generates the fitness of the individual according to the following function:

$$f(\mathbf{x}) = \sum_{t=1}^n r(\mathbf{x}, p_t) + s(\mathbf{x}, p_t) \cdot \beta, \quad (1)$$

where  $p_t$  is the position of the virtual rat at time step  $t$ . The term of reward ( $r(\mathbf{x}, p_t)$ ) may increase the fitness of the individual, following the rule:

$$r(\mathbf{x}, p_t) = \begin{cases} 1, & \text{if } p_t \text{ was not visited for the agent} \\ & \text{in the last } \gamma \text{ time steps} \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where  $\gamma$  is a parameter of the model that is related to the memory of the virtual rat.

On the other hand, the punishment ( $s(x, p_t)$  in Eq. 1) represents the exposure to the damage and decreases the fitness of the individual. It is known that the rat avoids the damage by spending more time in the enclosed arms, i.e., the level of damage is different for different positions of the maze. This way, the punishment is given by  $s(\mathbf{x}, p_t) \cdot \beta$  in Eq.(1), in which  $\beta$  is the weight of the punishment and

$$s(\mathbf{x}, p_t) = \begin{cases} -1, & \text{if } z_i < \alpha(p_t) \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where  $z$  is a random number, and  $\alpha(p_t)$  is the level of damage of the position  $p_t$  occupied in the time step  $t$ :

$$\alpha(p_t) = \begin{cases} \alpha_o \in [0, 1], & \text{if } p_t \text{ is an open arm} \\ \alpha_e \in [0, 1], & \text{if } p_t \text{ is an enclosed arm} \\ \alpha_c \in [0, 1], & \text{otherwise.} \end{cases} \quad (4)$$

The more the rat is exposed to danger, the higher is the chance of it being punished, which means that the punishment is probabilistic. That is, the fitness function is noisy and may be compared to a robot walking around an uncertain environment. The strategies described in the next section were studied to deal with this kind of uncertainty.

## Evaluation

As mentioned previously, we study four different evolutionary strategies for the GA. The individual's selection and fitness depend on the employed strategy. In the experiments, for each strategy, we calculate the mean over 30 executions of the GA and the population size is constant throughout the simulation (500 individuals).

The fitness of the individual is calculated based on the trajectory of the corresponding agent. After navigating in the EPM, the corresponding trajectory of the virtual robot is recorded. Then, the fitness based on this trajectory is sampled  $n$  times using Eq. (1) and the individual's fitness is given by the mean fitness of these  $n$  independent results. That is, the trajectory of a individual, which is the more costly part of the evaluation process, is performed only once. The mean of various fitness computations are calculated because of the uncertainties of the fitness function, as seen in the last section. The evaluation process may be seen in Figure 1. It is important to highlight that, in (Costa et al., 2012), the fitness is sampled only one time, i.e.,  $n = 1$ .

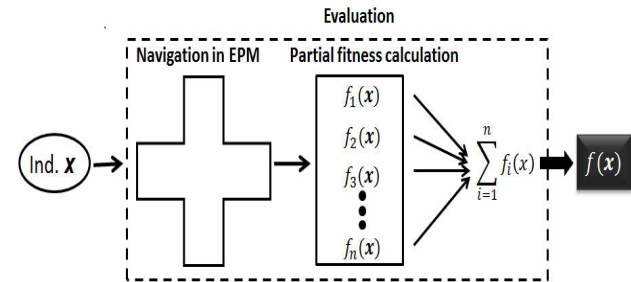


Figure 1: Process of individual evaluation. First, the individual navigates in a simulated five-minute test. The fitness based on this trajectory is then sampled  $n$  times. The resultant individual fitness is the mean of these independent fitness calculated.

## Strategies

The four strategies studied here are presented below.

**Strategy 1** The individuals are selected by elitism and tournament. By elitism, the two best individuals of the population are selected to the next population. In tournament selection, the best of two random individuals is selected, with probability 0.75, and the selected individual is then submitted to the reproduction operators. The tournament occurs until the population is completed. One-point crossover (with rate of 0.6) and mutation with uniform distribution (with rate of 0.05) are employed. The population evolves during 500 or 1500 generations (depending on the experiment), concluding an execution of the GA. Then (**after GA simulation**), the best individual of each of the 30 executions is evaluated with  $n$  partial fitness calculations and the mean fitness



represent the fitness of these individuals. It is important to observe that during the GA evaluation, each individual is sampled with  $n = 1$  on each generation of the evolutionary process.

**Strategy 2** This strategy is similar to Strategy 1. The difference is that each individual of the population is evaluated based on  $n$  samples of the fitness function. Then, at the end of each execution, the best individual is selected and it is not evaluated again.

**Strategy 3** Strategy 3 is the same of Strategy 1, except that the only type of selection is tournament. There is not elitism.

**Strategy 4** In this strategy we do the same of Strategy 2, but without elitism.

## Results

In previous experiments, we have tested several sets of parameters and the selected for the virtual robot simulations are:  $\gamma(p_t) = 3$ ,  $\beta = 5$ ,  $\alpha_o = 0.015$ ,  $\alpha_e = 0.012$  and  $\alpha_c = 0.011$ , which are used in all simulations presented here.

Table 1 exhibits the mean fitness obtained with the Strategies 1, 2, 3 and 4, based on 30 executions of 500 generations each one. In the evaluation, the fitness is computed 10 times for the best individuals of each run. By the table, it is clear that the best results are achieved with Strategy 1, in which the fitness exhibited corresponds to the mean of the best individuals of 30 executions of the GA evaluated after simulations. Strategies without elitism (Strategy 3 and 4) are worse than their similar with elitism. It shows that elitism is important to the model.

| Statistic | Strategy 1 | Strategy 2 | Strategy 3 | Strategy 4 |
|-----------|------------|------------|------------|------------|
| Mean      | -10.94     | -15.60     | -11.29     | -16.08     |
| Sd        | 3.98       | 1.31       | 4.52       | 0.01       |
| Min       | -18.84     | -16.12     | -22.48     | -16.11     |
| Median    | -11.15     | -16.072    | -11.60     | -16.08     |
| Max       | -2.88      | -11.41     | -1.98      | -16.06     |

Table 1: Simulated results of the fitness in 30 runs of 500 generations for Strategies 1, 2, 3 and 4, with  $n = 10$ .

To clarify how close our results are to the experiments with real rats, Figure 2 contains mean and standard deviation of the time spent in each arm and central position obtained by experiments with real and virtual robots. The experimental results with real rats are based on the trajectory of 47 rodents (Costa et al., 2012).

Virtual robots spend similar proportions of time in each arm of the EPM in relation to real rats, remaining substantially longer in the enclosed arms. The qualities of the strate-

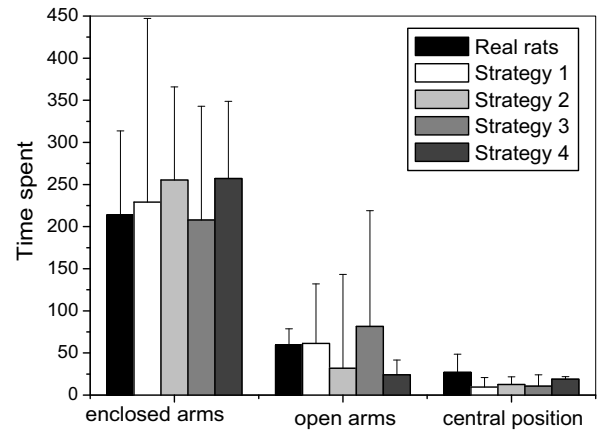


Figure 2: Mean and standard deviation of the time spent in enclosed and open arms and in the central position of the EPM for real rats and for virtual robots with the four strategies studied.

gies are in agreement with their respective fitness presented in Table 1.

Since we conclude that elitism is fundamental to keep best individuals in the population, the next results are all obtained with elitism. Our intention is to better understand the model and the influence of Strategies 1 and 2. For this purpose, we found the best individuals of 30 executions of both 500 and 1500 generations with Strategy 3 for  $n = 1$ ,  $n = 10$  and  $n = 100$ . Then, we evaluated them again after the runs, like is done in Strategy 1, calculating 100 partial fitness for each one of best individuals obtained in the 30 runs. We calculated the mean, standard deviation, median, maximum and minimum of the 30 fitness. The result is shown in Table 2.

| Statistic | 1ft 500ger | 1ft 1500ger | 10ft 500ger | 10ft 1500ger | 100ft 500ger | 100ft 1500ger |
|-----------|------------|-------------|-------------|--------------|--------------|---------------|
| Mean      | -9.79      | -9.46       | -11.50      | -7.97        | -0.92        | 1.60          |
| Sd        | 3.16       | 3.77        | 8.04        | 6.08         | 9.76         | 4.50          |
| Min       | -17.91     | -15.75      | -22.03      | -17.03       | -21.21       | -6.45         |
| Median    | -9.98      | -9.49       | -8.11       | -7.37        | 3.12         | 3.13          |
| Max       | 3.57       | 3.31        | 2.70        | -7.37        | 6.95         | 9.09          |

Table 2: Mean, standard deviation, minimum, median and maximum for the best individual of 30 executions of 500 and 1500 generations of Strategy 2 with  $n = 1$ ,  $n = 10$  and  $n = 100$ , associated with Strategy 1 with  $n = 100$ .

From Table 2, it is clear that increasing from 500 to 1,500 the number of generations in the GA, the quality of the re-

sults is improved in any studied situation. It also may be seen that increasing the number of samples of the fitness function is beneficial to the model. To improve the comprehension of what is happening, we analysed histograms of fitness coming from Strategy 2 with  $n = 1$  and  $n = 100$ . The best individual of the 30 executions is evaluated again with 1000 samples (partial fitness). These samples are in the histograms of Fig. 3, which also exposes the original fitness for  $n = 1$  and  $n = 100$ , i.e., the fitness of the best individual of the simulations whose trajectory was selected to be evaluated a thousand times ( $f_1$  and  $f_{100}$ , for the case with  $n = 1$  and  $n = 100$  respectively). We observe that  $f_{100}$  is a little higher than the average of 1,000 evaluations, while  $f_1$  is much higher, close to the maximum fitness obtained in the 1,000 evaluations. The value for  $f_1$  is high because the individual's fitness obtained with the calculation of only one sample is more susceptible to randomness. Since various individuals in the population have similar genome, that one with the maximum fitness is selected with higher probability, causing the bias in the histogram. On the other hand,  $f_{100}$  is the mean of a thousand samples, which ensures that the fitness will be near the mean fitness allowed for the trajectory evaluated for the various individuals with the same genome in the population. As this individual is selected with higher probability, it is progressively better along the executions. But this process occurs in a very slow way, compared to the case for  $f_1$ .

The Figure 3 enables to observe how noisy is the fitness function studied. One can observe, in the histograms, regions without samples. This occurs because the punishment weight is equal to 5.8 ( $\beta = 5.8$ ) in the runs, whereas the reward is equal to 1. So, there are values that are not covered by the possible combinations of rewards and punishments in Eq. 1 (one can remember that the reward is deterministic while the the punishment is probabilistic for a given trajectory).

Figure 4 shows the maximum, mean and minimum fitness of each generation for the best of 30 executions of the GA in simulations of Strategy 2 with  $n = 1$ ,  $n = 10$  and  $n = 100$ . As it was already presented, increasing the number of samples improves the fitness achieved by the optimisation process. The figure demonstrate that there is a jump in fitness around the 500th generation in strategy with  $n = 100$ , revealing that the virtual robot learns a new skill at this point of evolution. It is interesting to observe that it only happens when 100 samples of the fitness are considered. By analysing the amount of punishments and rewards received by the best individual of each generation, we can note that this jump is due to fact that the virtual robot have learned how to be more rewarded in his trajectory.

## Conclusions

The results show that explicit averaging is important to the studied problem, as our fitness function is probabilistic. The

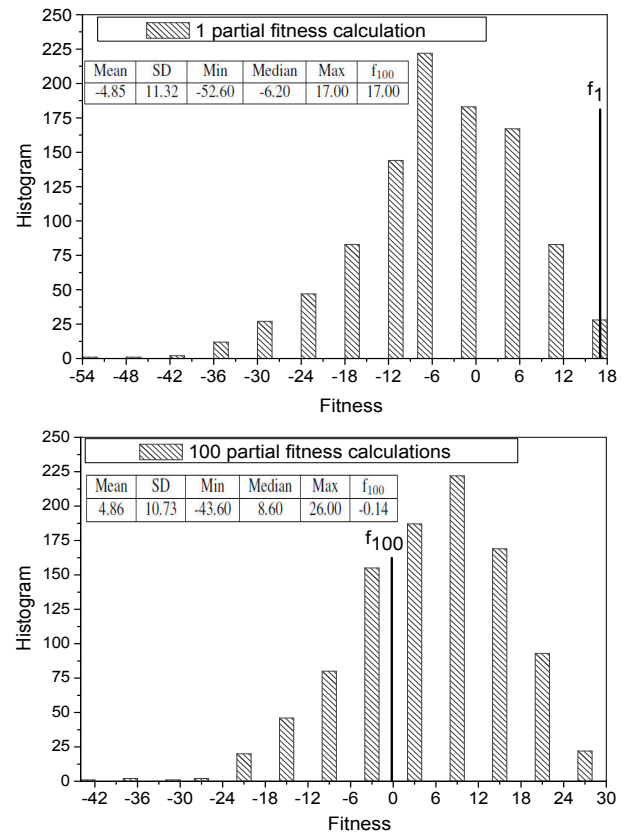


Figure 3: Histogram of 1,000 samples for the best individual obtained in 30 runs for Strategy 2 with  $n = 1$  (up) and  $n = 100$  (bottom). The tables show the mean, standard deviation, minimum, median, maximum and the original fitness.

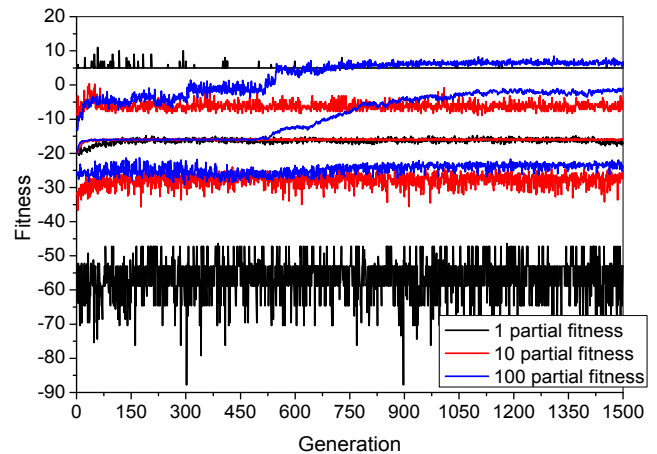


Figure 4: The up three lines are the maximum fitness; the central three lines correspond to the mean fitness; and the bottom lines are the minimum fitness of each generation for a run of Strategy 2 with  $n = 1$ ,  $n = 10$  and  $n = 100$ .

fitness function studied is very noisy, influenced by the reward and punishment weight values in the fitness function.

It is relevant to highlight that the navigation in the EPM is the most costly part of the evaluation process. It takes a robot a long time to perform its trajectory in a EPM's replica (as it did in (Shimo et al., 2010)). Hence, it is possible and fair to calculate a high number of samples of the fitness function for each individual evaluation.

Among the four strategies studied for the problem of the rat navigating in an elevated plus-maze, the most effective one is to evaluate the best individual with 100 samples of the fitness functions during the optimisation process. This is caused by the fact the explicit averaging smooths the effect of randomness in the fitness function. We also noticed that selection by elitism plays an important role in the evolution of the evolutionary robots for the problem studied here.

As a future work, we intend to study the influence performed by the ANN and each neuron from the hidden layer. Moreover, we intend to test the methods also in real robots.

## Acknowledgements

This research is supported by FAPESP, CNPq and CAPES. The authors are grateful to Helder Ken Shimo and to Julián Tejada for their contributions.

## References

- Baker, S. R., Bloom, N., and Davis, S. J. (2013). Measuring economic policy uncertainty. *Chicago Booth Research*, 13-02:1–47. Available at SSRN: <http://ssrn.com/abstract=2198490> or <http://dx.doi.org/10.2139/ssrn.2198490>.
- Buraimo, B. and Simmons, R. (2008). Do sports fans really value uncertainty of outcome? evidence from the english premier league. *International Journal of Sports Finance*, 3.3:146–155.
- Campi, M. C. and Calafiore, G. (2004). Decision making in a uncertain environment:: the scenario-based optimization approach. *Gloria-Mundi*, pages 1–14.
- Carmignani, F. (2003). Political instability, uncertainty and economics. *Journal of Economic Surveys*, 17(1):1–54.
- Cioffi-Revilla, C. (1998). *Politics and Uncertainty: Theory, Models and Applications*. Cambridge University Press.
- Costa, A. A., Roque, A. C., Morato, S., and Tins, R. (2012). *Lecture Notes in Computer Science. 1a ed, v. 7435*, chapter Model Based on Genetic Algorithm for Investigation of the Behavior of Rats in the Elevated Plus-Maze, pages 151–158. Berlin: Springer Berlin Heidelberg.
- Cyert, R. and March, J. (1963). *A Behavioral Theory of the Firm*. NJ: Prentice-Hall.
- Duncan, R. B. (1972). Characteristics of organizational environments and perceived environmental uncertainty. *Administrative Sciences*, 17:31–327.
- Emery, F. and Trist, E. (1965). The causal texture of organizational environments. *Human Relations*, 18(1):49–63.
- Fitzpatrick, J. M. and Grefenstette, J. J. (1988). Genetic algorithms in noisy environments. *Machine Learning*, 3(2-3):101–120.
- Forrest, D. and Simmons, R. (2002). Outcome uncertainty and attendance demand in sport: The case of english soccer. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 51(2):229–241.
- Galbraith, J. (1973). *Designing complex organizations*. Reading, Mass.: Addison-Wesley.
- Giddings, J. M. (2002). Modeling the behavior of rats in an elevated plus-maze. bachelors thesis. Technical report, Acadia University.
- Gil, R. (2008). Revenue sharing distortions and vertical integration in the movie industry. *The Journal of Law, Economics, & Organization*, 25(2):579–610.
- Goschin, S., Littman, M. L., and Ackley, D. H. (2011). The effects of selection on noisy fitness optimization. *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 2059–2066.
- Goulas, E. and Zervoyianni, A. (2013). Economic growth and crime: does uncertainty matter? *Applied Economics Letters*, 20:420–427.
- Graeff, F. G. (1990). *Handbook of anxiety, vol 3. The neurobiology of anxiety*, chapter Brain defense systems and anxiety. Amsterdam: Elsevier Science Publishers.
- Jennet, N. (1984). Attendances, uncertainty of outcome and policy in scottish league football. *Scottish Journal of Political Economy*, 31(2):176–198.
- Jin, Y. and Branke, J. (2005). Evolutionary optimization in uncertain environments - a survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317.
- Knowles, G., Sherony, K., and Hauptert, M. (1992). The demand for major league baseball: A test of the uncertainty of outcome hypothesis. *The American Economist*, 36(2):72–80.
- Lattimore, P. and Witte, A. (1986). *The Reasoning Criminal: Rational Choice Perspectives on Offending*, chapter Models of Decision Making Under Uncertainty: The Criminal Choice, pages 129–155. NY: Springer-Verlag.
- Lawrence, P. R. and Lorsch, J. W. (1967). *Organization and environment: Managing Differentiation and Integration*. Boston: Harvard University Press.
- Marris, P. (1996). *The Politics of Uncertainty: Attachment in Private and Public Life*. Routledge.
- Miller, B. L. and Goldberg, D. E. (1996). Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation*, 4:113–131.
- Miller, D. and Shamsie, J. (1999). Strategic responses to three kinds of uncertainty: Product line simplicity at the hollywood film studios. *Journal of Management*, 25(1):97–116.
- Miranda, D. A., Conde, C. A., Celis, C. C., and Corzo, S. P. (2009). Modelado del comportamiento de ratas en laberinto en cruz elevado basado en redes neuronales artificiales. *Revista Colombiana de Física*, 41(2):406–408.

- Montgomery, K. C. (1955). The relation between fear induced by novel stimulation and exploratory behavior. *Journal of Comparative and Physiological Psychology*, 48:254–260.
- Naceur Jabnoun, A. K. and Yusuf, A. (2003). Environmental uncertainty, strategic orientation, and quality management: A contingency model. *Quality Management Journal*, 10(4):17–31.
- Peel, D. and Dennis, T. (1992). The demand for football: Some evidence on outcome uncertainty. *Empirical Economics*, 17(2):323–331.
- Pindyck, R. S. (2007). Uncertainty in environmental economics. *Review of Environmental Economics and Policy*, 1(1):45–65.
- Ratray, M. and Shapiro, J. (1996). Noisy fitness evaluation in genetic algorithms and the dynamics of learning.
- Salum, C., da Silva, A. C. R., and Morato, S. (2000). Anxiety-like behavior in rats: a computational model. *Neural Networks*, 13(1):21–29.
- Salum, C., da Silva, A. C. R., and Morato, S. (2003). Conflict as a determinant of rat behavior in three types of elevated plus-maze. *Behavioural Processes*, 63:87–93.
- Sano, Y. (2002). Optimization of noisy fitness functions by means of genetic algorithms using history of search with test of estimation. *Proceedings of the 2002 Congress on Evolutionary Computation*, pages 360–365.
- Shimo, H. K., Tejada, J., Roque, A. C., and Tinós, R. (2010). *Use of evolutionary robots as an auxiliary tool for developing behavioral models of rats in an elevated plus-maze*. In: *Proceedings of the 2010 Eleventh Brazilian Symposium on Neural Networks*.
- Sondahl, F. and Stonedahl, S. H. (2010). Heuristics for sampling repetitions in noisy landscapes with fitness caching. *Genetic and Evolutionary Computation Conference, GECCO 2010*, pages 273–280.
- Vany, A. D. (2004). *Hollywood Economics - How Extreme Uncertainty Shapes the Film Industry*. London/ NY: Routledge.
- Walf, A. A. and Frye, C. A. (2007). The use of the elevated plus maze as an assay of anxiety-related behavior in rodents. *Nature Protocols*, 2:322–328.



# Speciation Dynamics: Generating Selective Pressure Towards Diversity

Heiko Hamann

Department of Computer Science, University of Paderborn, Paderborn, Germany  
heiko.hamann@uni-paderborn.de

## Abstract

Recent approaches in evolutionary robotics (ER) propose to generate behavioral diversity in order to evolve desired behaviors more easily. These approaches require the definition of a behavioral distance which often includes task-specific features and hence a priori knowledge. Alternative methods, that do not explicitly force selective pressure towards diversity (SPTD) but still generate it, are known from the field of artificial life such as artificial ecologies (AE). In this study, we investigate how SPTD is generated without task-specific behavioral features or other forms of a priori knowledge and detect how methods of generating SPTD can be transferred from the domain of AE to ER. A promising finding is that in both types of systems, in systems from ER that generate behavioral diversity and also in the investigated speciation model, selective pressure is generated towards unpopulated regions of search space. We conclude by hypothesizing how knowledge about self-organizing SPTD in AE could be transferred to the domain of ER.

## Introduction

Methods of evolutionary computation have been successful as optimization technique for many years. Also the optimization of behaviors, which can justifiably be called ‘generation of behaviors’, in the field of ER (Nolfi and Floreano, 2000) has proven to be effective. However, the next step in this research towards more complex behaviors and tasks seems to be particularly difficult. Such a complex task could involve, for example, several successive sub-tasks whereas learned earlier sub-tasks have no utility before later sub-tasks are learned as well. The relative simplicity of investigated tasks in ER, especially when compared to natural systems, is, for example, discussed by Nelson et al. (2009).

Evolving robot behaviors becomes even more challenging if the necessary a priori knowledge is minimized which is necessary to achieve generally applicable approaches. Notably this concerns the fitness function and how elaborated it is. Nelson et al. (Nelson et al., 2009) define several fitness function classes such as the

behavioral fitness functions which incorporate a lot of a priori knowledge (fitness function ‘selects for behavioral features of a presupposed solution to a given task’, i.e. *how* the task is accomplished) and the aggregate fitness functions which incorporate a very low degree of a priori knowledge (fitness function measures *what* the robot has accomplished and not *how* it was accomplished). Behavioral fitness functions are applied in ER because otherwise behaviors of certain complexities cannot be evolved with a reasonable commitment of resources.

An option is to increase the diversity in the population, e.g. by fitness sharing, see (Sareni and Krähenbühl, 1998). However, these methods include the measurement of a distance between genotypes which is computationally intractable for common encodings in ER, such as artificial neural networks (ANN) (Mouret and Doncieux, 2012). Instead promising recent results suggest to increase the behavioral diversity during the search or within the current population and measure the distances between behaviors which can be done efficiently. Examples are novelty search (Lehman and Stanley, 2011) and the approach by Mouret and Doncieux (Mouret and Doncieux, 2009, 2012). Novelty search (Lehman and Stanley, 2011) operates without an actual objective function. Instead selective pressure is generated towards behaviors that have not been seen before in the evolutionary run. A desired behavior is that that maximizes the behavioral distance to all known behaviors. For illustration and later use this is shown schematically in Fig. 1a. The circles represent behaviors that were found during the evolutionary run (graded colors represent different generations) and their position in search space. Behaviors that are close to known behaviors are undesirable which is represented by low values of the fitness function  $F$  around circles. Consequently selective pressure towards unpopulated regions in search space is generated which is represented accordingly by steep, upward slopes around circles. Multi-objective behavior diversity (MOBD, Mouret and Doncieux, 2009) includes the

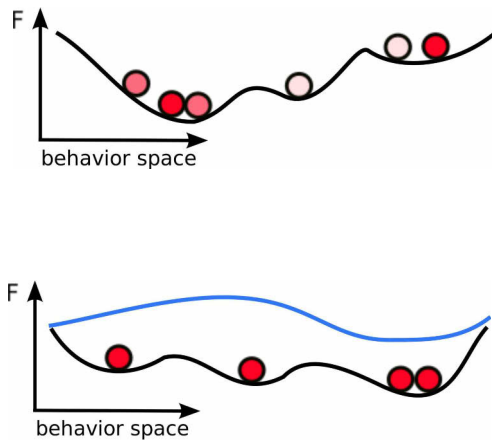


Figure 1: Schematic representation of the fitness functions generated by novelty search (Lehman and Stanley, 2011) and MOBD (Mouret and Doncieux, 2009); circles represent known behaviors, selective pressure is towards bigger values of  $F$ .

behavioral distance only as a component in the multi-objective fitness function. In contrast to novelty search it only accounts for the behaviors in the current population. This is sketched in Fig. 1b.

For approaches of behavioral diversity a distance between behaviors needs to be defined, which typically will be task-specific. Hence, this is a process similar to defining an appropriate fitness function for a given task in standard ER. It is argued that even naive behavioral distance definitions can improve the evolutionary process, such as definitions based on the final position of the robot or of movable objects at the end of the evaluation (Mouret and Doncieux, 2009; Lehman and Stanley, 2008). However, if we follow the analogy to fitness functions, it seems likely that for more complex tasks the measure of behavioral distance will also need to be more complex. This factual connection gets even more plausible if we reconsider the above mentioned comment by Nelson et al. (2009) about behavioral fitness functions (which select for behavioral features). In the design of the behavioral distance measure not only *what* is accomplished is relevant but also *how* it is accomplished. In a simple exploration task this is not really apparent, if the behavioral distance is defined on the robot's final position (Mouret and Doncieux, 2009), but it addresses at least the order of how the maze is explored. In a more complex task such as the locomotion of a biped robot (Lehman and Stanley, 2011) it gets more apparent

as the behavioral distance is defined by the trajectory of the center of mass (how) instead of just the total offset (what). Hence we state the hypothesis that also methods based on behavioral distances will run into similar problems as seen in fitness function design such as having to define 'task-specific hand-formulated functions that contain various types of selection metrics' (Nelson et al., 2009) and consequently having to involve a high degree of a priori knowledge.

Based on our hypothesis we assume that neither genotypic distances nor behavioral distances are able to generate sustainable diversity in ER. What could be candidate solutions? Natural evolution represents a perfect standard for the generation of diversity. Particularly we are interested in evolutionary radiation which is an increase in taxonomic diversity. A typical example of a radiation is the Cambrian explosion. It generated a diversity that is comprehensible when looking at the corresponding phylogenetic tree which gets literally bushy within a comparatively short period of time. Each branching corresponds to an event of speciation. Hence, to understand diversity we want to understand speciation and the underlying process that generates SPTD. Our objectives are 1) to detect how SPTD is generated in a self-organizing system (e.g., speciation) and 2) how to transfer this knowledge to ER.

Next we summarize the knowledge on how speciation operates. Coyne and Orr (2004) ask: 'Why are there species?' and they comment: 'we regard it as one of the most important unanswered questions in evolutionary biology'. Accordingly they do not answer but only discuss the question. They point to Maynard Smith and Szathmary (1998) who consider three explanations. One, species are discrete 'stable states' formed by a self-organizing system. However, this option lacks a mechanism that would explain the origin of species. Two, species fill discrete ecological niches. Third, reproductive isolation is an inevitable result of evolutionary divergence. The latter two are dependent on reproductive isolation and are not mutually exclusive (Coyne and Orr, 2004). A conclusive concept that connects both is that of 'adaptive peaks' (Dobzhansky, 1951). Species are adaptive peaks that are separated by 'adaptive valleys' which are genotypes that are unfit for survival. Whether these adaptive valleys are due to ecological or environmental effects is kept open. Another aspect is whether asexual or sexual reproduction is considered. While in the case of asexual reproduction ecological niches seem to be indispensable to create a reproductive barrier, in the case of sexual reproduction the reproductive barrier might be generated without explicit ecological niches especially because sexual selection could be effective. To summarize we note that evolutionary radiation and speciation are yet difficult to understand as

also pointed out by Venditti et al. (2010): ‘Attempts to understand species-radiations [...] should look to the size of the catalogue of potential causes of speciation shared by a group of closely related organisms rather than to how those causes combine.’ The conclusion concerning inspirations for new methods to generate diversity in artificial evolution, especially ER, regrettably has to stay inconclusive for now due to the limited knowledge about the natural system itself. However, it is the starting point of the following investigations.

Instead of detecting methods to generate speciation, what are prerequisites for evolutionary radiation, that is, for active speciation? A prerequisite could be the existence of a complex environment that favors the formation of adaptive peaks and that initiates specialization in the organisms. Another prerequisite could be the existence of an ecology which creates complexity in the interaction of different organisms and species. The suitability of providing a complex environment within applications of ER is limited, notably if a complex environment is not part of the desired task. On the contrary, a considerable amount of research on the creation of AE in order to evolve behaviors has been reported. For example, the minimal ecology, that of just two species, is investigated in studies of coevolution (Floreano and Nolfi, 1997) and AE, possibly with many species, are popular in the field of artificial life (Ray, 1991). The studies of coevolution have an emphasis on the actually evolved behaviors combined with considerations about their utility and complexity. It turns out that ‘the co-evolutionary process tends to fail into dynamical attractors in which the same solutions are adopted by both populations over and over’ (Nolfi and Floreano, 1998b). In AE studies the actual behaviors are of less interest, instead the evolutionary process as a whole is usually investigated in more detail. In addition it is also unclear how these AE would have to be designed to generate desired behaviors for a given task. Common to both is their high sensitivity to parameters and the challenge of creating actively progressing evolutionary processes. However, they are interesting examples of how SPTD is generated in a self-organizing process without incorporating a priori knowledge about how a certain behavior is accomplished. Summing up, the prerequisite to generate diversity is a minimum of complexity that provokes the emergence of adaptive peaks. The trigger could be ecological or environmental features but also effects of sexual selection. The goal of this study is 1) to detect and measure SPTD which is generated by a self-organizing process in AE, and 2) to determine how methods from the domain of AE could be transferred to ER. To the knowledge of the author, published studies on behavioral diversity tend to focus either on self-organizing diversity without aiming for the solution of

| parameter                   | FAM     | EAM        |
|-----------------------------|---------|------------|
| max. age                    | 4 years | 4 years    |
| max. energy                 | 2.0     | 2.0        |
| max. energy per seed        | 2.0     | 2.0        |
| search energy cost          | 0.1     | 0.1        |
| max. num. male mating       | 5       | 5          |
| max. num. generations       | 1000    | 1000       |
| initial number of birds     | 400     | 150        |
| initial beak size mean      | 5.5     | 5.5        |
| initial beak size variance  | 0.5     | 3.5        |
| beak size interval          | [1, 10] | [1, 10]    |
| dry season length           | 43 days | 30 days    |
| initial number of seeds     | 5000    | 6300       |
| feeding square size         | 10      | 10         |
| world size                  | 100     | 100        |
| variance of offspring prop. | 0.03    | 0.01       |
| assortative range $\Delta$  | 0.5     | [0.01, 10] |
| seed width $W$              | n.a.    | [0.01, 10] |

Table 1: Parameter settings for fixed assortative mating (FAM) and for evolved assortative mating (EAM); some parameters differ from (Woehrer et al., 2012).

a given task (e.g., Ray (1991)) or on explicitly imposed SPTD while searching for the solution of a particular task (e.g., Lehman and Stanley (2011)). In the following we investigate a speciation model of Woehrer et al. (2012) and an extension to it in order to investigate how SPTD is generated in this self-organizing system. The results are then compared to approaches that explicitly impose SPTD.

## A model of speciation

One distinguishes several types of speciation, such as allopatric speciation and sympatric speciation. Previously it was thought that speciation happens mostly allopatrically, that is, by spatial separation of populations which then develop reproductive isolation. More recent results (Coyne and Orr, 2004) support that sympatric speciation, which is speciation within the same geographic region, might be more common than expected. Here, sympatric speciation is of more interest because it is a self-organizing, evolutionary process while allopatric speciation occurs due to external forces (arguably except for migration). In terms of the application of speciation to increase the behavioral diversity in ER, sympatric speciation is preferred as it does not need a priori knowledge while allopatric speciation would need an implementation of a cause.

Woehrer et al. (2012) report an artificial life model of sympatric speciation based on sexual selection, in particular assortative mating which is a mating pattern where mating between individuals with similar geno-

types or phenotypes is more likely. The model is inspired by the natural system of finches on the Galápagos Islands. Although an island setting might let allopatric speciation appear as a good and exclusive explanation this does not seem to be the case on the Galápagos Islands (Woehrer et al., 2012). Woehrer et al. (2012) point to the specialty of the proposed system that combines natural selection and sexual selection acting on the same trait which is directly related to so-called ‘magic traits’ (Servedio et al., 2011). Here we reproduce their results, report an extension of the model, and perform additional measurements in simulations.

The artificial system models a bird population of dynamic size. A bird is modeled by age, beak size, energy level, and gender. To survive the birds have to forage (implemented as random search) for seeds. These are modeled by energy, location, uniformly distributed size, and distributed in discrete space of size 100 units  $\times$  100 units (see table 1 for used parameter settings). Selective pressure is imposed by the limited resource of seeds. Initially of the dry season a number of seeds is placed in the world which decreases consecutively over a period of 30 or 43 simulated days (depending on which setting is used) as the birds forage from it. The birds’ search for seeds is limited by their beak size  $s$  because they can only feed on seeds of size  $[s - 1, s + 1]$ . The search costs energy on each day and fed seeds add energy to the bird’s energy. If a bird runs out of energy during the dry season it dies. Those that survive may attempt to reproduce. Reproduction is based on sexual selection and assortative mating (with random mating no speciation was observed (Woehrer et al., 2012)). Females with beak size  $s_f$  select only mates with beak size  $s_m$  within the female’s beak size interval  $s_m \in [s_f - \Delta, s_f + \Delta]$  for assortative range fixed to  $\Delta = 0.5$ . The offspring has a beak size averaged over its parents plus Gaussian noise and random gender. For all remaining details please see (Woehrer et al., 2012) and table 1.

A typical run is shown in Fig. 2a which is a plot of all beak sizes that occur in the population over 375 generations. The resemblance to a phylogenetic tree is obvious. Also the drift of species, branching into two species, and the extinction of species can be noticed. Hence the model of Woehrer et al. (2012) is a simple model of self-organized speciation and can be used as an easy-to-handle analogy to the studies on behavioral diversity in ER. The interval of allowed beak sizes  $[1, 10]$  is the equivalence to the behavior space and a bird’s beak size would be the 1-d equivalence to the behavior defined by ANN. The extreme difference between high-dimensional ANN and the simplistic 1-d beak size interval is not a limiting factor of this analogy because the speciation model possesses the one qualitative feature that is relevant for this study, namely self-organized generation

of diversity. The assortative mating corresponds to allowing recombination only for ANN that share a ‘magic trait’ which is defined by sexual selection and could be a behavioral feature. In addition, this way the emergence of speciation relies crucially on a pre-defined parameter (assortative range  $\Delta$ ). In order to avoid such a pre-defined measure for the above mentioned reasons we can allow the evolutionary algorithm to vary features of sexual selection. The perfect solution would be to evolve the full process of sexual selection which is, however, beyond the scope of this paper. Instead we restrict the following investigations to the evolution of the allowed difference between females’ and males’ beak size defined by the assortative range  $\Delta$  in this particular case of assortative mating. While this solves the problem of having pre-defined parameters, this would still correspond to a pre-defined process of sexual selection. That way we are able to investigate whether the evolutionary process can self-organize towards a higher degree of diversity without being forced to do so by a parameter setting.

It turns out that the model is very sensitive to settings of the assortative range  $\Delta$  which is typical for such systems as pointed out above (Sec. ). If  $\Delta$  is set too low, species extend over a narrow interval of beak sizes, feed from a small set of seeds that they are able to eat, and become extinct often (data not shown). If  $\Delta$  is set too big, no speciation is observed because one big connected component of birds in ‘beak size space’ emerges. Still we proceed and allow the assortative range  $\Delta$  to be evolved. The extension of the above model is described in the following. The parameter  $\Delta$  that defines sexual selection by setting the beak size range is defined now as an individual bird property. It is passed on by an average over the parents plus Gaussian noise (see table 1 for parameters). Furthermore, we introduce an evolved parameter of each individual bird called ‘seed width’  $W$  to increase the attractiveness of being a specialist. In addition to the beak size it also determines the interval of seed sizes a bird is able to feed on ( $[s - W, s + W]$ , priority is with the more restrictive interval) and the energy of a seed is scaled by  $1/W^2$  (for  $W < 1$  a seed’s energy is increased quadratically in  $W$ ). The system shows also speciation without this additional feature of seed width  $W$  but the inclusion of seed width stimulates speciation (data not shown).

For the following presentation and analysis of our results we define a technical concept of species in this simple model. In order to do so we interpret the distribution of a population’s beak sizes as a graph whereas each bird’s beak size represents a node and two such nodes are connected to each other if they are within each others individual assortative range of sexual selection defined by  $\Delta$ . A species is defined by the graph-



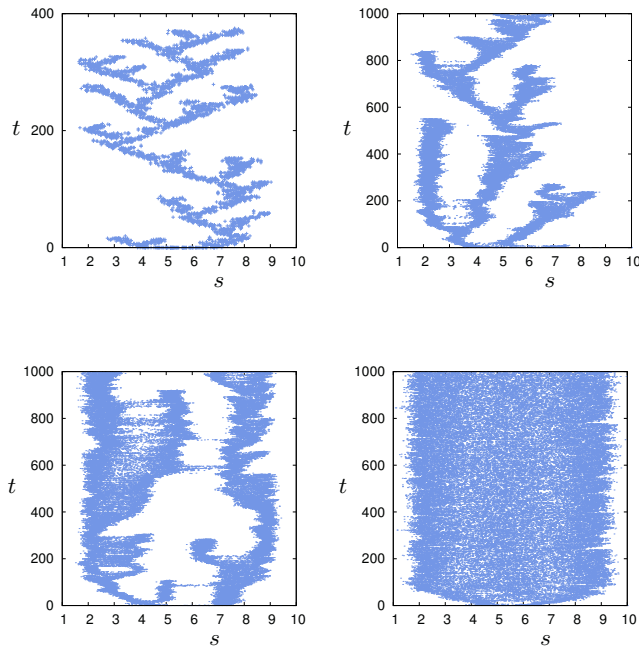


Figure 2: Beak sizes  $s$  over generations  $t$  for fixed and evolved assortative mating; for parameters see Tab. 1; sub-figures b-d differ only in random initialization.

theoretic concept of connected components: a subset of nodes and for each pair of such nodes there is a path connecting both. With this definition we are able to implement an automatic classifier that determines where and how many species exist in a given configuration.

## Results

In the following we investigate whether speciation is observed, we investigate the distribution of beak sizes over time to find what is beneficial for speciation in this extended model, the distribution of branch lengths, and the dynamics of species. With this extended model we obtain for different random initializations results (for parameters see Tab. 1) that are characterized by three classes: showing speciation (see Fig. 2b), intermediate (see Fig. 2c), and not showing speciation (see Fig. 2d). Following our connected-component definition, species within limited beak-size intervals and clear separation are noticed in Fig. 2b. In Fig. 2c species are not clearly separated at all times. For example, at  $t \approx 600$  one species spans almost over the whole beak-size interval. In Fig. 2d, a single species covers the whole interval. Hence this system does not reliably self-organize towards diversity, at least for the tested parameters.

For the following investigations we classify occurrent

configurations. It turns out that four configurations are frequent: only species in the left part of the beak size interval ( $s < 4.5$ , called *left*, frequency: 3.24%), only species in the right part ( $s > 6.5$ , called *right*, frequency: 3.14%), species distributed over the whole interval (called *all-over*, frequency: 88.37%), and species in the two outer parts but not in the middle (called *symmetrical*, frequency: 5.24%). Concerning the evolved assortative range  $\Delta$ , the all-over configuration is distinguishable from the three other configurations. In Fig. 3a the distributions of all occurring assortative ranges over a number of evolutionary runs are compared. The mean for all-over configurations is 1.9 and bigger than those of the others (about 1.3). With bigger assortative range a species spans over big intervals more easily. Consequently big assortative ranges counter diversity in terms of speciation. The populations that spread over the whole interval are bigger than those showing diversity (mean of about 200 birds compared to about 100) because they exploit the energy provided by seeds fully. Consequently they are less prone to fluctuations and have a smaller risk of extinction. At the same time they make sure to exploit seeds of all sizes. Hence the low-diversity solution actually seems to be the evolutionary more robust approach which raises the question of how speciation could be stimulated additionally (an optional target of an investigation which is beyond this paper would be the tradeoff between a generalist's advantages and costs due to seed width  $W$ ). In turn it is possible to force the system into speciation by forcing bi- or multi-modal distributions of seed sizes (Woehrer et al., 2012). However, this is tweaking the environment which is not a good option for our application in ER.

Generally, populations with big assortative ranges seem to be more stable and there is also a tendency of spreading over the whole interval with increasing time. We are able to support this claim by investigating the distribution of branch lengths in the phylogenetic trees. The branch length is the time period between split-ups of species and/or their extinction. Based on an automatic check for connected components, species and the corresponding branch lengths are determined in the implementation of the model. The distribution of branch lengths based on independent evolutionary runs ( $4.4 \times 10^5$  samples) is shown in Fig. 3b (squares). In the analysis of this data we follow Venditti et al. (Venditti et al., 2010) who discuss and interpret different branch length distributions of phylogenetic trees. While they apply their methods to natural systems we found that they can also be applied to this artificial system. The best fit we found is the Weibull distribution ( $ca/bx^{(a-1)} \exp(-(x/b)^a)$ ,  $a = 0.32$ ,  $b = 0.065$ ,  $c = 29 \times 10^3$ ) which Venditti et al. (2010) interpret in the following way: “the Weibull density can accommodate

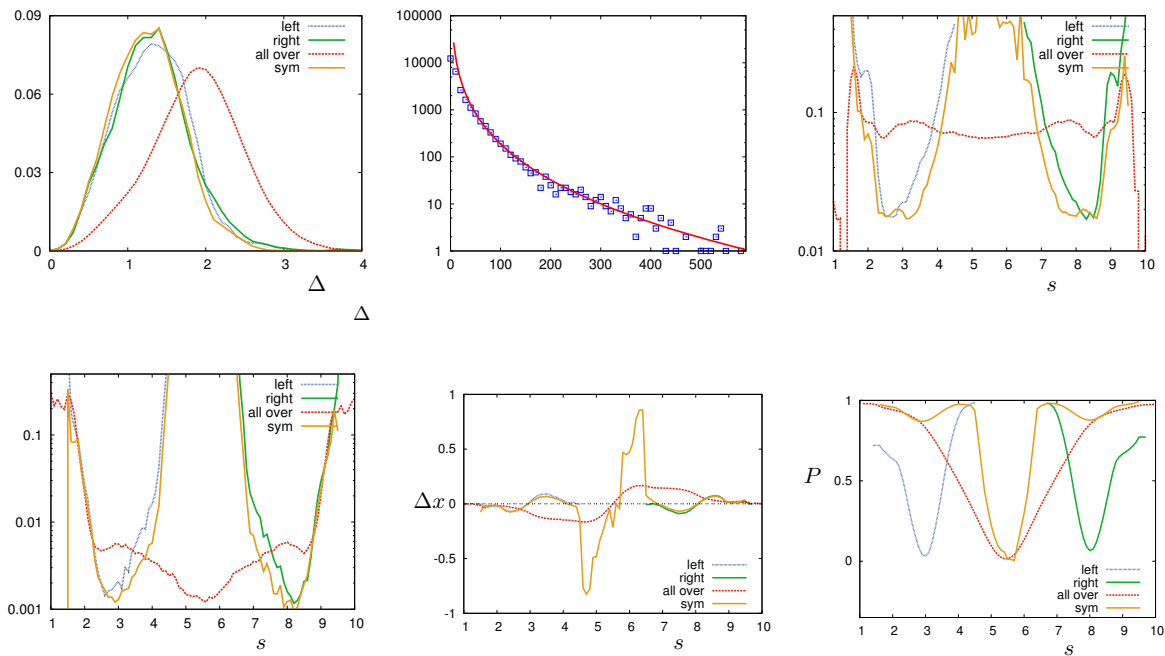


Figure 3: Distribution of the assortative range  $\Delta$  for different configurations and the distribution of branch lengths in the phylogenetic tree with fitted Weibull distribution; birth and death rate of species over beak size for different configurations; averaged movement of species and implied potentials over beak size for different configurations.

the probability of speciation changing according to the amount of divergence from the ancestral species. This model will fit the data if, for example, species are either more or less likely to speciate the older they get.” This supports our finding that old species tend to speciate less as they tend to span the whole interval.

Despite these shortcomings in the sustainability of the evolution of species we are able to investigate the dynamics of speciation in this system. Our aim is to detect and measure how SPTD is generated without explicitly forcing it. The evolution of species in space and time based on the above connected-component definition can be interpreted as a discrete birth–death process combined with drifting motion. For the above defined four classes of configurations the birth and death rate depending on the beak size were measured, see Fig. 3c and d (qualitative, no errors shown). The birth rate is bigger by about an order of magnitude because in our implementation the merger of species was not classified as death; only actual extinction was classified as death. Interestingly the birth rate for the all-over configuration is almost homogeneous while the other configurations have a dip at  $s \approx 3$  and  $s \approx 8$ . This is most likely because species in the all-over configuration are in average almost evenly distributed while in the other configurations species are more likely to be positioned at  $s \approx 3$  and/or  $s \approx 8$ . Once this ‘niche’ is covered, a birth at

the same position is not possible. The death rates of all configurations have peaks at the bounds which is explained by the smaller number of available seeds once birds cover also seed sizes that do not occur ( $< 1$  or  $> 10$ ). Interestingly beak sizes keeping a distance to the bounds support survival for all configurations.

In the following we want to measure the dynamics of species, that is, the average movement of species in beak size space in certain configurations. A measurement of how the species drift for the four configurations is shown in Fig. 3e.  $\Delta x$  gives the average displacement of a species from one to the next generation for  $1.9 \times 10^7$  samples. Positive values  $\Delta x > 0$  describe the drift of a species towards bigger beak sizes and negative values  $\Delta x < 0$  describe drift towards smaller beak sizes. For the configurations *left*, *right*, and *all-over* the average motion of species indicates spreading and species keep moving towards the bounds even when already approaching them which means they move into regions of high death rates (Fig. 3d). The configuration *symmetrical* is different because it has a stabilizing effect at  $s \approx 4$  and  $s \approx 6.5$  which are maxima. This is, however, a direct effect of the classification and due to situations when an *all-over* configuration turns into a *symmetrical* configuration. Still it is valid data within our classification scheme. To draw a direct connection to applications in ER we determine the potentials that are implied

by the average movement of species (for example, similar to gravitational potentials) which is here merely an integration over the average movement. Fig. 3f shows the 4 potentials of the 4 configurations (normalized to similar scales). These potentials are important findings for this study because they are emergent fitness functions with selective pressure towards bigger values in the same way as in the above schematic representations of the fitness functions for novelty search and MOBD (see Fig. 1). Similarly also here currently populated regions in beak-size space are less desirable and there is pressure towards unpopulated regions. For example, the potential for configuration *left* has a minimum at  $s = 3$  which corresponds to the typically populated position in this configuration as determined by low birth and death rates around  $s = 3$  (Fig. 3c and d) and an average motion of  $\Delta x = 0$  (Fig. 3e). These potentials are the confirmation that this self-organizing system certainly generates SPTD. While, for example, in novelty search that pressure is explicitly enforced by pushing towards behavioral diversity, here the selective pressure is a feature of the ecological system. Unpopulated regions in beak-size space correspond to big resources of energy in the form of seeds no one forages for. Birds that manage to push into these regions compete with few fellows, gather plenty of energy, and increase their fitness for survival. This analogy shows on the one hand that approaches based on behavioral distance should not just be considered as engineered abstractions but rather as bio-inspired approaches that have a direct connection to ecological features of speciation. On the other hand it shows an option of how to generate a self-organizing SPTD as discussed in the following.

## Discussion and Conclusion

In this paper we have detected self-organizing generation of SPTD in an AE and we have motivated the need for methods that generate diversity, particularly behavioral diversity in the field of ER. Standard methods to increase the diversity in a population but also recent methods based on behavioral distances include a considerable amount of a priori knowledge because they (tend to) include behavioral features in the definition of the behavioral distance that measure *how* the robot accomplishes a task. Mouret and Doncieux (2012) discuss the analogy between fitness function design and behavioral distance measure design: “More importantly, novelty search critically depends on a good behavior characterization to create a gradient. Researchers in ER used to craft fitness function to create a perfect fitness gradient; novelty search users have to craft the behavior distance to create a similar gradient. This last option may be easier for some problems but eventually some distances will be hard to define.” Hence,

even with methods not relying on fitness functions the former problem of including a priori knowledge about the task persists. Mouret and Doncieux (Mouret and Doncieux, 2012) discuss how the ideal setting for ER might look like: “In an ideal ER setup, ER researchers would only define a high-level fitness function and let the generic evolutionary process do the rest. This goal could be achieved with a generic behavioral distance function that could be used with most ER tasks while still improving the evolutionary process.” Whether such a generic measure of behavioral distances can be found is unknown. Instead we study diversity in natural evolution that is generated in a self-organizing process and our focus is on diversity by speciation. Due to limited biological knowledge about why species exist it is also difficult to determine prerequisites for the generation of diversity in artificial systems. Therefore we investigated the simple speciation model by Woehrer et al. (2012) and extended it to allow the evolution of features that define the process of sexual selection based on assortative mating. Our findings indicate that self-organizing speciation is achievable but the system is sensitive to parameter settings as known from other AE.

It seems difficult to provoke evolutionary dynamics that favor diversity over uniform solutions without imposing diversity by predefined environmental features (e.g., multi-modal distributions of seeds, see (Woehrer et al., 2012)). In fact it seems questionable whether sustainable generation of diversity is possible without stimulating influences from a dynamic environment or a complex ecology with intensive inter-species interaction. A beneficial finding is the analogy between methods based on behavioral distances in ER and the investigated self-organizing ecology in terms of the selective pressure that is generated. In both systems selective pressure is generated towards unpopulated regions of the search space (cf. Figs. 1 and 3f). Hence we follow that both systems relate to each other and that there might be a way of transferring the methods of generating selective pressure in AE to ER.

In the speciation model this selective pressure is an ecological feature because unpopulated regions of search space hold plenty of seeds which increase the fitness for survival once they are foraged. While this is easily implemented in this speciation model by defining a beak-size search space, it is unknown how distributing seeds in search space transfers as analogy to the behavior space of ER. This triggers an important research question: How to define AE in the context of ER that generate SPTD without explicitly addressing particular task-specific behavioral features and without including a priori knowledge. This could be done by a data structure that covers the full behavior space and keeps track of which regions of this abstract space have been vis-

ited (following the analogy: regions where most of the seeds have been eaten up). However, this would be most likely intractable and probably also task-specific. Solutions for this dimensionality reduction problem based on heuristics would possibly have similar difficulties as, for example, function approximation in reinforcement learning (Sutton and Barto, 1998). Presumably any method that maps behaviors from the actual behavior space into a smaller feature space will suffer from being either task-specific or of limited benefit although very simple mappings were shown to be beneficial in simple tasks (Mouret and Doncieux, 2009; Lehman and Stanley, 2008). Seemingly the statistics about the frequencies of behaviors is embedded into the environment in natural systems similarly to the concept of stigmergy in swarm intelligence. A candidate solution would be to evolve behaviors in an embodied system which allows for embedding behavioral statistics in the environment. Fortunately, an embodied approach is feasible in ER (Bredeche et al., 2012; Stradner et al., 2012).

Furthermore we have reported measurements (Figs. 3c-f) that allow for modeling speciation as a discrete birth-death process combined with a spatial feature determined by drifting motion towards unpopulated regions. That way speciation truly is linked “to rare stochastic events that cause reproductive isolation” (Venditti et al., 2010). In addition we have shown that the analysis of branch length distributions by Venditti et al. (2010) is also applicable in artificial systems and might prove to be instrumental in classifying artificial systems of speciation. In future work we plan to continue investigations of how to integrate AE into behavior space that is not task-specific and that generates SPTD in complex tasks for ER. In addition it might be desirable to investigate also models of speciation that evolve themselves features of sexual selection that generate diversity.

## References

- Bredeche, N., Montanier, J.-M., Liu, W., and Winfield, A. F. (2012). Environment-driven distributed evolutionary adaptation in a population of autonomous robotic agents. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):101–129.
- Coyne, J. A. and Orr, H. A. (2004). *Speciation*. Sinauer Associates.
- Dobzhansky, T. (1951). *Genetics and the origin of species*. Columbia University Press.
- Floreano, D. and Nolfi, S. (1997). Adaptive behavior in competing co-evolving species. In *Proceedings of the fourth European Conference on Artificial Life*, pages 378–387. MIT Press.
- Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In Bullock, S., Noble, J., Watson, R., and Bedau, M. A., editors, *Artificial Life XI: Proc. of the 11th Intern. Conf. on the Simulation and Synthesis of Living Systems*, pages 329–336. MIT Press.
- Lehman, J. and Stanley, K. O. (2011). Improving evolvability through novelty search and self-adaptation. In *Proceedings of the 2011 IEEE Congress on Evolutionary Computation (CEC’11)*, pages 2693–2700. IEEE.
- Mouret, J.-B. and Doncieux, S. (2009). Using behavioral exploration objectives to solve deceptive problems in neuro-evolution. In *Proceedings of the 11th annual conference on Genetic and Evolutionary Computation (GECCO’09)*, pages 627–634. ACM.
- Mouret, J.-B. and Doncieux, S. (2012). Encouraging behavioral diversity in evolutionary robotics: an empirical study. *Evolutionary Computation*, 20(1):91–133.
- Nelson, A. L., Barlow, G. J., and Doitsidis, L. (2009). Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Syst.*, 57:345–370.
- Nolfi, S. and Floreano, D. (1998b). How co-evolution can enhance the adaptive power of artificial evolution: Implications for evolutionary robotics. In Husbands, P. and Meyer, J.-A., editors, *Evolutionary Robotics: 1st European Workshop*, vol. 1468 of *LNCS*, pages 22–38.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press.
- Ray, T. S. (1991). Evolution and optimization of digital organisms. In *Proc. of the 1990 IBM Supercomputing Competition*, pages 489–531. The Baldwin Press.
- Sareni, B. and Krähenbühl, L. (1998). Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3):97–106.
- Servedio, M. R., Doorn, G. S. V., Kopp, M., Frame, A. M., and Nosil, P. (2011). Magic traits in speciation: ‘magic’ but not rare? *Trends in Ecology and Evolution*, 26(8):389–397.
- Maynard Smith, J. and Szathmáry, E. (1998). *The Major Transitions in Evolution*. Oxford University Press.
- Stradner, J., Hamann, H., Zahadat, P., Schmickl, T., and Crailsheim, K. (2012). On-line, on-board evolution of reaction-diffusion control for self-adaptation. In Adami, C., Bryson, D. M., Ofria, C., and Pennock, R. T., editors, *Alife XIII*, pages 597–598. MIT Press.
- Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.
- Venditti, C., Meade, A., and Pagel, M. (2010). Phylogenies reveal new interpretation of speciation and the Red Queen. *Nature*, 463:349–352.
- Woehrer, M., Hougen, D., and Schlupp, I. (2012). Sexual selection, resource distribution, and population size in synthetic sympatric speciation. In Adami, C., Bryson, D. M., Ofria, C., and Pennock, R. T., editors, *Proc. of the 13th Intern. Conf. on the Simulation and Synthesis of Living Syst. (Alife’13)*, pages 137–144. MIT Press.



# Multi-Crease Self-Folding by Uniform Heating

Shuhei Miyashita and Daniela Rus

Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, USA  
shuheim@csail.mit.edu

## Abstract

This paper presents a self-folding method for multi-crease structures. The proposed method utilizes the symmetric breaking of a 3-layered two-dimensional sheet, where an inner contraction sheet induces shear force when heated, which directs the inclined folding direction. The fabrication technique developed enables distant placements of tiling patterns of the surfaces. The experimental result shows that by applying uniform heat, a feature with 62 folds can be simultaneously folded, an advantage over manual folding. The method presented is a new instant fabrication technique for making semi-rigid structures.

## Introduction

Biological entities achieve their fertile morphogenetic processes based on protein self-assembly and self-folding. Inspired by such processes, various attempts at the automated structuring of robot bodies have been made in the field of robotics. Hawkes et al. (2010) demonstrated self-folded ship and plane origami structures employing shape-memory alloy (SMA) actuators. Cheung et al. (2011) showed arbitrary 3D shapes that could theoretically be folded from a single strand. The underlying principle behind these approaches is assigning an individual actuator (i.e., shape memory alloy, or an electric motor) to each folding site. Thus, scalability of the system remains an open challenge. Recently, Onal et al. (2011) showed that the body of an insect-shaped, legged robot could be folded from a single laser-cut polyester sheet. MEMS technology employing a pop-up technique was proposed by Whitney et al. (2011). Felton et al. (2013) advanced Onal's model and incorporated the self-folding of a robot's body by assigning conductors around the hinges for localized heating by current application.

## Self-Folding Method

In this study, we use uniform heating to attain self-folding. We utilize thermal deformation of a contraction sheet (polyvinyl chloride, PVC) sandwiched by rigid structural layers with different gap widths, such that the shear force of a contraction layer induces a bending motion of the surface. We show the basic principle of the folding method in Fig. 1. Schematics of the fold are shown in side views on the left side in the figures, with corresponding snapshots on the right. The method is capable of (1) simultaneous mountain and valley

folds, (2) simultaneous multiple folds, and (3) coarse angle control by varying gap widths.

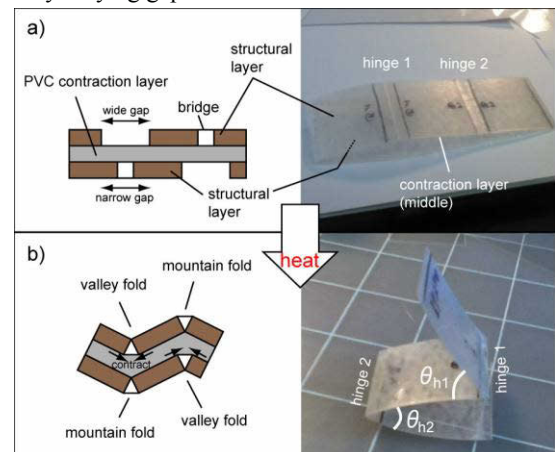


Fig. 1. Self-folding process by uniform heating.

## Accurate Folding Angle by Angle Fold

Attaining an arbitrary folding angle of a sheet structure is difficult at any scale, for it is dependent on the folding torque generated by the material. In order to attain accurate arbitrary folding angles, we focus on the kinematics of angle folds, specifically, the V-fold, in which one of the angles (termed output angle  $\theta_{out}$ ) can be precisely controlled by being kinematically coupled with another actuable angle (termed input angle  $\theta_{in}$ ). Since the derivative of  $\theta_{out}$  is small when  $\theta_{in}$  is small, in spite of the absolute change, the idea here is to “crudely” actuate  $\theta_{in}$  to exercise precise angle control over  $\theta_{out}$ . The concept for V-folds is illustrated in Fig. 2.

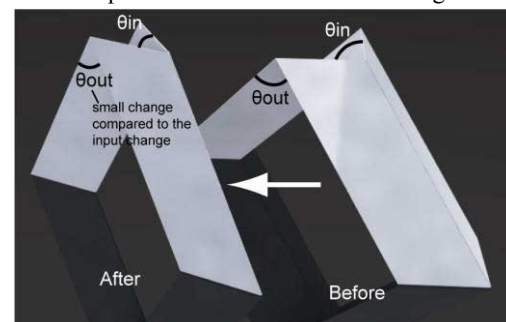


Fig. 2. Input angle and output angle in V-fold.

## Fabrication

Since the pattern of a structural layer involves multiple discrete tiles (islands), structural layers are cut out and placed on a semi-rigid backing layer. Once a laser makes the pattern, parts that position at gaps are manually peeled off. A contraction layer is then inserted and sandwiched by the backing-laminated structural layers, by folding it in half. Finally, the backing layer is removed from the structure, and the desired self-folding sheet is obtained.

## Result

The temperature control for the self-folding process is managed in an oven. To realize ideal uniform heating for the body, the sheet is hung from the ceiling. To demonstrate the advantage of self-folds, we chose an origami pattern that could only be folded if all the creases were folded simultaneously. This structure, which is shown in Fig. 4, consists of 62 mountain and valley folds.

Fig. 3 shows the self-folding process by uniform heating. Starting at room temperature and ramped up to 65 degrees, the process is complete in about 5 minutes.

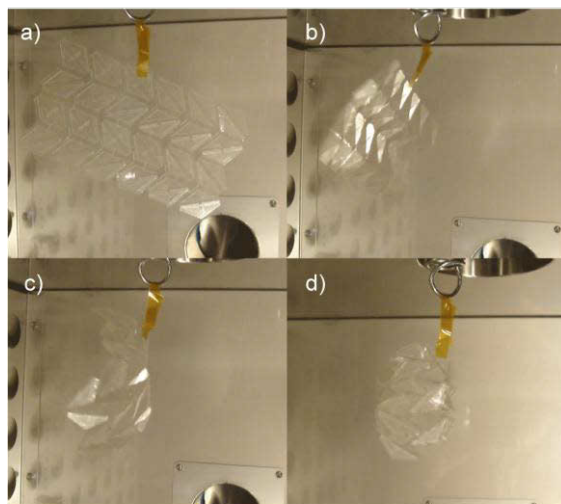


Fig. 3. Self-folding process by “baking.”

The fabricated structure is shown in Fig. 4. The process, although applicable one time only, is reliable and fast. The method is suitable for types of fold that consist of many pleat patterns, because heat is applied to the entirety of the targeted material.

## Conclusion

This work presents the self-folding of a multi-crease structure by uniform heating. We first developed a technique that, by having different gap widths between a contraction layer, achieved the simultaneous folding of mountain and valley folds by heating the middle contraction layer. We further developed a fabrication technique, designing the island

features and placing them onto the contraction sheet. The self-folding process achieved is fast and reliable, and is promising for the fabrication of more complex structures.



Fig. 4. Self-folded structure, which consists of 62 mountain and valley folds.

## References

- Hawkes, E., An, B., Benbernou, N. M., Tanaka, H., Kim, S., Demaine, E. D., Rus, D., and Wood, R. J. (2010) Programmable matter by folding. *Proceedings of the National Academy of Sciences*, vol. 107, no. 28, pp. 12 441–12 445.
- Cheung, K. C., Demaine, E. D., Bachrach, J. R., and Griffith, S. (2011) Programmable assembly with universally foldable strings (moteins). *IEEE Transactions on Robotics*, vol. 27, pp. 718–729.
- Onal, C. D., Wood, R. J., and Rus, D. (2011) Towards printable robotics: Origami-inspired planar fabrication of three-dimensional mechanisms. *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4608–4613.
- Whitney, K. M. J.P., Sreetharan, P.S., and Wood, R. (2011) Pop-up book mems. *Journal of Micromechanics and Microengineering*, vol. 21, no. 11, p. 115021.
- Felton, S. M., Tolley, M. T., Onal, C. D., Rus, D., and Wood, R. J. (2013) Robot self-assembly by folding: A printed inchworm robot. *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, accepted.
- Hunt, G. W. and Ario, I. (2005) Twist buckling and the foldable cylinder: an exercise in origami. *International Journal of Non-Linear Mechanics*, vol. 40, pp. 833–843.
- Min, C. C. and Suzuki, H. (2008) Geometrical properties of paper spring. *Manufacturing Systems and Technologies for the New Frontier*, pp. 159–162.

## Artificial Reaction Network Agents

Claire E. Gerrard<sup>1</sup>, John McCall<sup>1</sup>, Christopher Macleod<sup>1</sup>, and George M. Coghill<sup>2</sup>

<sup>1</sup>IDEAS Research Institute,  
Robert Gordon University,  
Aberdeen, Scotland, UK

<sup>2</sup>Department of Computing Science,  
University of Aberdeen,  
Aberdeen, UK, Scotland

c.e.gerrard@rgu.ac.uk, j.mccall@rgu.ac.uk, g.coghill@abdn.ac.uk, chris.macleod@rgu.ac.uk

### Abstract

The Artificial Reaction Network (ARN) is an Artificial Chemistry representation inspired by cell signaling networks. The ARN has previously been applied to the simulation of biological signaling pathways and to the control of limbed robots. In this paper we create multiple cell-like autonomous agents using ARN networks. It is shown that these agents can simulate some aspects of the behavior of biological amoebae. To demonstrate practical applications of such agents they are then reconfigured as a swarm of robots in a simulated oil spill clean-up operation. We demonstrate that ARN agents, like amoebae, can autonomously recognize environmental patterns and produce emergent behavior. The results show that such agents may be useful in biological simulation and furthermore may have practical applications in swarm robotics.

### Introduction

Unicellular organisms have evolved an astonishing array of complex behaviors. Some can avoid light with photo-sensitive spots; some actively hunt prey; while others can build protective shelters (Ford, 2009). It has been shown that single cells achieve such primitive intelligence by storing and processing information through the complex dynamics of interacting chemicals (Bray, 1995; Arkin and Ross, 1994). Within a cell, data is represented by a set of spatially distributed concentrations of chemical species; the instantaneous set of which corresponds to the cell's current state. Intricate networks of chemical reactions termed cell signaling networks (CSNs), process this information by transforming input species into output species. In this way, cells are able to respond to changes within their environment, communicate with other cells, and perform internal self maintenance operations. Several researchers highlight the processing capabilities of CSNs and their similarities to Artificial Neural Networks (ANNs) (Bray, 1995; Bhalla, 2003). For example, it has been demonstrated that a network of such reactions can perform Boolean and Fuzzy Logic functions and are equivalent to a Turing machine (Bray, 1995; Arkin and Ross, 1994). Furthermore, CSNs contain features such as feedback loops and interconnectivity, thus forming

highly complex systems (Bray, 1995; Bhalla, 2003). It is possible to exploit computational features of such chemical processing to create an Artificial Chemistry (AC). In its broadest sense, an AC describes a man-made system which is similar to a real chemical system (Dittrich, et al., 2001). The Artificial Reaction Network is an example of an AC and is based on properties and mechanisms found in CSNs. In our previous work, it was applied to simulate the chemotaxis signaling pathway of *Escherichia coli*, and later investigated as a means to produce complex temporal waveforms to control limbed robots (Gerrard, et al., 2011; 2012a; b).

In this paper, a single ARN network is instantiated and used as the internal control system for multiple instances of cell-like autonomous distributed agents. Our first objective is to show that ARN agents have application in the simulation of biological cells, their interactions, and resulting emergent behaviors. This is addressed by using the agents to simulate aggregating cells of the slime mould *Dictyostelium discoideum* and comparing the emergent behaviors with the literature. Our second objective is to show that by reconfiguring the inputs to each agent's ARN, the same agents can produce other distinct behaviors. Our final objective is to show that ARNs have application as the control systems for distributed robotic agents within real world environments. Here, we apply the agents to the task of cleaning up a simulated oil spill within a simplified search environment.

The paper is structured as follows: the first section provides an overview of the ARN representation. This is followed by an overview of the operation of the ARN agents. The experimental details and results are presented next; these are followed by the conclusions.

### The Artificial Reaction Network

In this section we provide a brief summary of the ARN representation. A full account can be found in our previous work (Gerrard, et al., 2011; 2012a; b).

The ARN comprises a set of networked reaction nodes (circles), pools (squares), and inputs (triangles) as shown in

figure 1. Each pool stores the current available chemical species concentration (avail); this concentration represents data within the system. Thus, the complete set of pool concentrations at time  $t$ , corresponds to the system's current state. Inputs are a special type of pool, the only difference being that they are not updated by flux at each time step, and are used to represent continuous concentrations, for example, environmental inputs or enzymes. Each circle corresponds to a reaction unit, representing a reaction between a number of chemicals. Data is processed by reaction nodes transforming incoming pool values to connected outgoing pool values. Connections symbolize the flow of chemical into and out of reaction units and their weight ( $w$ ) corresponds to reaction order. Connections provide the facility to create complex control structures using combinations of inhibitory and excitatory connections.

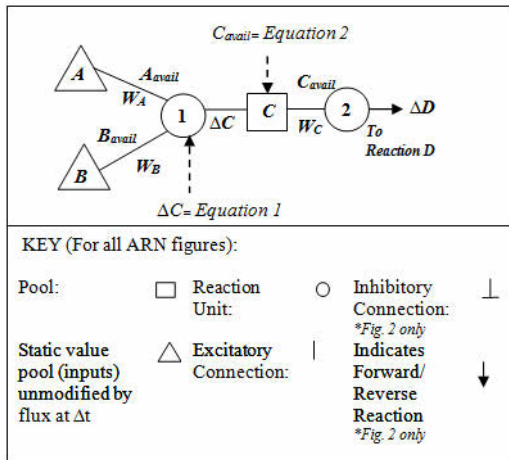


Figure 1: The Artificial Reaction Network representation.

$$\Delta C = \left( K_f(C) \left( A_{avail}^{W_A} B_{avail}^{W_B} \right) - K_r(C) \left( C_{avail}^{W_C} \right) \right) \Delta t \quad (1)$$

$$C_{avail} = C_{avail} + \left( \Delta C - \frac{W_C}{W_C + \alpha} \Delta D \right) \quad (2)$$

Where:

A, B, C, D = Species Concentrations

W = Reaction order (weight)

avail = Available species concentration

$K_f$  = Forward rate constant

$\Delta C$  = Change in species concentration C

$K_r$  = Reverse rate constant

$\alpha$  = sum of other incoming weights

Figure 1 shows the reaction between species A and B to produce species C. At time interval  $\Delta t$ , each reaction unit's temporal flux value is calculated by applying Euler's approximation to the differential rate equation given in (1). This value is then used to update the current concentration of each reaction's connecting pools as shown in (2). Pools may asymptotically approach 0, and thus below a particular threshold a pool is considered empty and its value set to zero. A reaction step may proceed if it meets its preconditions. Preconditions are met if incoming inhibitory pools are

inactive, and incoming excitatory pools are active. Similarly a reaction step will fulfill a number of post conditions: participating reactants are consumed and products generated—the amount of which will depend on the parameters of the reaction step.

## ARN Agents

This section describes the behavioral modes of each agent and the structure and operation of the ARN network controlling them. In the experiments outlined in this paper, a number of autonomous ARN controlled software agents termed "Cytobots" ("cyto" from Greek for cell, and "bot" from robot) are initialized and move around asynchronously within a 2D simulated environment containing a distribution of artificial chemicals. The artificial chemicals represent attractants of either food or cAMP (cyclic adenosine monophosphate). When an agent moves to a new position, the surrounding level of chemical is used to set the inputs to its ARN. Consequently this changes the internal state of the ARN and updates the agent's trajectory. During this process, the agent modifies the state of the environment by, for example, consuming food or releasing cAMP. Similar to the way in which a CSN acts as the control system to a cell, the behavior of each cytobot is controlled by its own instance of an ARN network. The ARN network architecture is based on a combination of functional structural motifs found in actual biochemical networks (Tyson and Novak, 2010). Each ARN instance is updated asynchronously with all other instances. In this way, each instance directs an agents' movement asynchronously to other agents, enables it to react to situated environmental patterns, and allows it to stigmergically communicate with other cytobots to contribute to higher level function.

The cytobot ARN network was designed to produce two simple behavioral modes: foraging and starvation, both are based on the movement patterns of single celled organisms as described in the following sections. The cytobot ARN is composed of 6 subnetworks as shown in figure 2. Each subnetwork contributes a functional aspect to either or both starvation and foraging behaviors. The subnetworks are discussed in the following sections.

### Cytobot Foraging Behavior

Cytobots forage by performing a biased random walk. This pattern of movement is exemplified by the bacteria *E. coli*, where foraging cells alternate periods of runs (forward motion) and tumbles (random redirections). By comparing concentrations of attractants and repellants in a temporal fashion, the organism is able to reduce the frequency of tumbles up concentration gradients of attractants, and down gradients of repellants, resulting in overall travel to more favorable conditions (Vladimirov and Sourjik, 2009). In the foraging mode a cytobot performs a similar random biased walk movement pattern. At each new position X, an agent redirects to a new random angle between 0 and 360 degrees (tumble). The agent then moves forward in a straight line for a number of time steps based on the level of detected food at position X (run). The cytobot consumes food (if present) at each passing location.



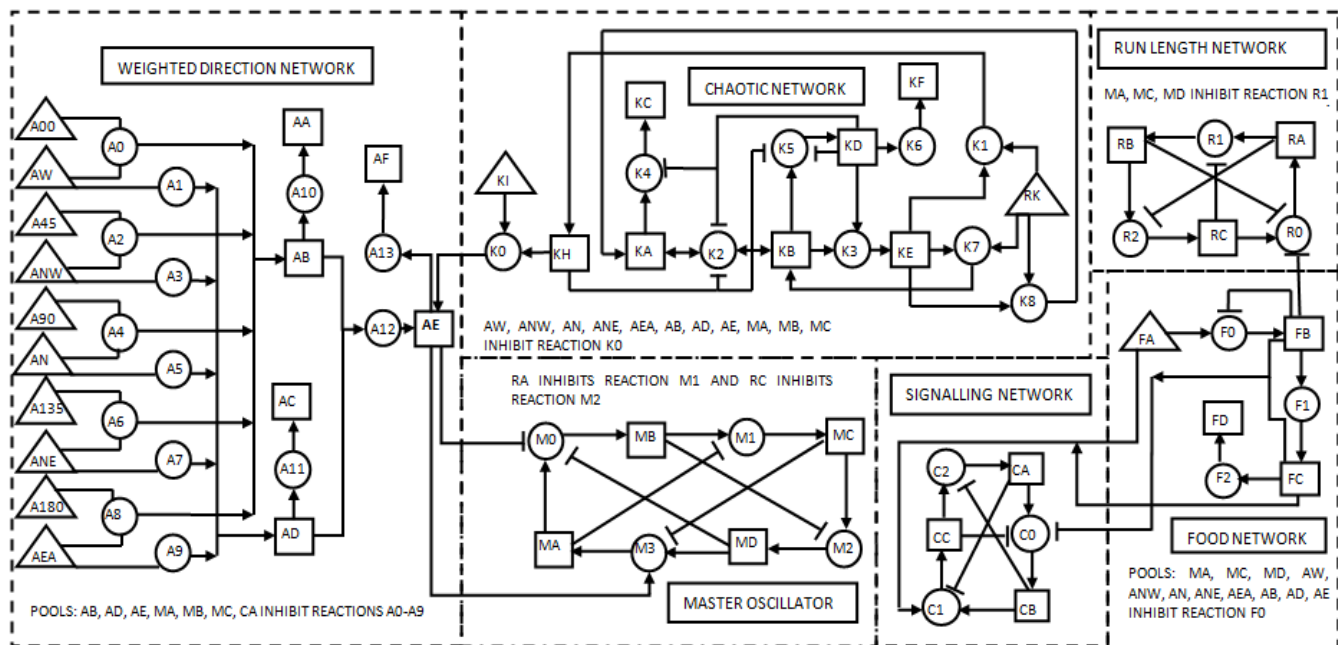


Figure 2: The cytobot ARN network comprising 6 subnetworks. Each cytobot is controlled by an instance of this network.

### Cytobot Starvation Behavior

The starvation behavior is based on the pattern of motion displayed by starving cells of the cellular slime mould *D. discoideum*. During the organisms' vegetative stage, cells move up gradients of folic acid secreted by its bacterial prey. When the food resource has been depleted, the amoebae begin to starve and enter the aggregation phase of their life cycle. During aggregation, starving cells secrete cAMP which serves as a signal to attract surrounding amoebae towards a central location (McCann, et al., 2010). During the aggregation phase, *D. discoideum* cells are polarized, thus one side becomes the leading edge which always faces in the direction of travel (McCann, et al., 2010). Depending on parameters such as environmental conditions, and the cell population density, migrating cells often form transient emergent patterns such as streams, waves and spirals (McCann, et al., 2010; Dallon and Othmer, 1997). Streaming describes a pattern of motion where cells line up in close order files, with the head of one following the rear of another (McCann, et al., 2010).

In these experiments the agents enter starvation mode if food has not been consumed within a time period. Here, instead of turning in a random direction, the new direction is weighted toward the highest concentration of cAMP within its surrounding area. As discussed later, by representing the external chemicals in different ways within the simulated environment, different high level behaviors can be produced by the agents.

### The Master Oscillator

The master oscillator network (see figure 2) functions to synchronize all the outputs from all the other subnetworks together and is what each agent references at each time step to ascertain its current behavior. It is a simple closed loop, with

a token unit of chemical cycling around it. It consists of 4 reaction units: M0, M1, M2, and M3 (all with reaction rate of 1) and 4 pools MA, MB, MC and MD. Each pool activates one of three behaviors, and for every time step that a particular pool contains the token unit, its corresponding behavior is performed. Pool MA activates turn, MC activates run and pools MB and MD activate stop. If these pools were switches to motor actuators on a simple wheeled robot, pool MC would switch on all wheel motors, while pool MA would switch on wheel motors on the left side only, thus turning the robot. The remaining pools would act as off switches. The other subnetworks inhibit or excite the reaction units of the master oscillator to allow or prevent chemical flow. The number of time steps that a chemical is present in a particular pool indicates the length of time that a particular behavior is performed. Thus if pool MC contains a chemical for 10 time steps, then the agent will move forward for 10 time steps; similarly if this were pool MA, the agent would turn for 10 time steps.

### The Food Network and the Run Length Network

The food network senses the level of food within the environment and connects to the run length network to modify the number of steps forward according to the level of food sensed. The value of food at a cytobot's current position is stored at input pool FA. The forward rate of reaction node F0 is 1, thus all food is transferred to pool FB in a single time step. The presence of chemical in pool FB inhibits the run network reaction R0 for a number of time steps according to the level of food (by setting forward rate of unit F1 to 1 and weight to 0 this can be an exact correlation). This in turn stops pool RC in the run length network from emptying. Pool RC inhibits reaction M2 of the master oscillator thus preventing pool MC from emptying for the same number of time steps.

As discussed previously, the number of time steps which pool MC contains the token unit represents the number of time steps to move forward.

### The Signaling Network

The signaling network functions as a switch between starvation and foraging mode. Low food levels trigger the starvation response and allow the weighted direction network to control each new angle. Sufficient food will switch off the weighted direction network and allow the chaotic network to control each new angle. It is a simple closed loop with a token unit of chemical flowing around it. Pool CA acts as a switch between foraging and starvation behavior, where the presence of chemical in CA inhibits the weighted direction network; while its absence switches on the weighted direction network; this in turn inhibits the chaotic network, as shown in figure 2. In this component, all reaction units have a forward flux of 0.5; which ensures a minimum number of time steps for each behavior.

### The Weighted Direction Network

The weighted direction network senses cAMP within the agent's immediate environment and calculates a tumble angle which is weighted toward higher cAMP levels. This network interfaces with the environment via a number of receptor pools (AW, ANW, AN, ANE, AEA) which sense the level of cAMP around the cytotob. These pools represent receptors positioned at fixed points around the front of its perimeter. Limiting the signal detection to one side facilitates representation of polarization in *D. discoideum*, where that side becomes the leading edge. For each receptor input pool, there is a static pool containing a fixed level of chemical which represents the angle of the receptor relative to the cytotob. Directions start from AW (west) with a corresponding numeric value of 0 (A00) and progress in 45 degree steps through each direction to east (thus maximum value is 180). Detected signals are classed as being in one of the following cardinal/ordinal directions: W, NW, N, NE, and E. Thus signals are detected from all directions above its horizontal plane. All connections have a weight of 1 with the exception of the connection between pool AD and reaction A12 which has a weight of -1. This negative connection raises the sum of chemical detected in pool AD to -1, which multiplied by AB, allows the average angle to be calculated. The calculated angle interfaces with the remaining subnetworks at pool AE. In an actual organism, receptors are set around the cell perimeter and direct movement appropriately.

$$h \equiv ((n - 90) + c) \bmod 360 \quad (3)$$

Where:

h= new heading (relative to external frame)

n = count of time steps pool MA contained chemical

c = current heading (relative to the external frame)

In this simulation, for simplicity, a count of the number of time steps that MA contains the token unit is processed to

gain the turn angle using (3). Thus if the number time steps is 120 and the agent is facing north, then the current heading would equal 0 (relative to the external frame) and the new heading would equal 30.

### The Chaotic Network

The chaotic network, as shown in figure 2, is responsible for generating pseudo random angles which agents use to perform the foraging tumble behavior. It is a networked implementation of a Logistic Map, see (4). Ulam and von Neumann (1947) were the first to examine a Logistic Map as a pseudo random number generator and it has been successfully used in this capacity by several researchers (Patidar, et al., 2009). The probability density distribution of the Logistic Map is non-uniform and is given in (5).

$$X_{n+1} = \lambda X_n(1 - X_n) \quad (4)$$

Where:

$X_n$ = state variable of value  $0 \leq X_n \leq 1$

$\lambda$ = system parameter of value  $1 \leq \lambda \leq 4$

$$P(X) = \frac{1}{\pi \sqrt{X(1-X)}} \quad (5)$$

Where:

P(X) = probability of X occurring

At the start of the simulation, pools KA and KB of each cytotob's chaotic network are initialized to the same random value between 0 and 1 (to 5 decimal places). This value represents the first value of X (where X represents the state variable of (4)). All the other pools are initialized to 0 with the exception of the static pools KI and RK whose initial values are 360 and 1 respectively. Reaction K2 is responsible for generating each new value of X and has a forward and reverse rate of 4 (the logistic map exhibits chaotic behavior when  $\lambda$  is 4). The connection between KA and K2 has a weight of 1 and the connection between K2 and KB has a weight of 2. The remaining series of reactions function to copy the value of X 3 times, where 2 copies serve as the new initial values of KA and KB and the remaining copy participates in the final output of the network at KH. Static pool KI has a fixed value of 360 which in reaction K0, allows the network to convert the pseudo random number at KH to an angle value between 0 and 360. However, reaction K0 cannot proceed until all 11 pools that inhibit it are empty. These inhibitory connections ensure that random angles are not output while the agent is in starvation mode, and that pool AE is empty before adding more chemical.

## Slime Mould Aggregation Simulation

In the following experiments cytotobs are used to model the behavior of aggregating *D. discoideum* cells, where each cytotob represents a cell. In each experiment the emergent patterns, numbers of mounds, and length of time to mound formation is examined. A total of 10 experiments are performed at varying population densities of cytotobs (p) and

different ranges of detection of cAMP ( $r$ ), as shown in table 1. The environment contains no food, thus each agent immediately enters and stays in the previously described starvation mode. The agents' behavior is initially explored at biologically realistic  $p$  and  $r$  values and compared with the behavior of the actual organism and other simulations. These parameters are then extended into ranges outwith the biological range in order to examine the emergent properties of the system.

```

FOR each cytotobot
  Get current agents' facing direction  $C_F$ 
  Assign a value to direction  $C_F$  using statement 1

  FOR each (index  $n$ ) detected cAMP signal
    Get detected signal incoming direction  $C_A$ 
    Assign a value to direction  $C_A$  using statement 1
    IF  $C_A = C_F$  THEN  $k_n = 3$ 
    ELSE IF  $C_A = C_F - 1$  OR  $C_A = C_F + 1$  THEN  $k_n = 2$ 
    ELSE IF  $C_A = C_F - 2$  OR  $C_A = C_F + 2$  THEN  $k_n = 1$ 
    ELSE  $k_n = 0$ 
    END IF
    Calculate distance  $d_n$ 
    Store each  $C_A$  with  $k_n$  and  $d_n$ 
  END FOR

  Calculate  $W_A$  for current agent using Equation 6
END FOR

Statement 1: East = 1; North East = 2; North = 3; North West = 4; West = 5

Where:
 $W_A$  = total weight of direction A
 $N$  = total number of agents within range of detection
 $d_n$  = distance of current agent from agent  $n$ 
 $C_A$  = direction of incoming signal detected by current agent
 $C_F$  = the current agents facing direction
 $k_n$  = value of cAMP signal from agent  $n$ 

```

Figure 3: Pseudocode to calculate the strength of detected cAMP at each direction relative to the cell.

$$W_A = \sum_{n=1}^N \frac{k_n}{d_n} \quad (6)$$

Cytobots move within a simulated 2D environment of area  $5.06 \text{ mm}^2$ - approximately half the maximum recorded aggregation territory reported in the literature (Dallon and Othmer, 1997). Each pixel represents  $4.5 \text{ }\mu\text{m}$  and the grid is  $500 \times 500$  pixels, giving a total area of  $5.06 \text{ mm}^2$ . In nature, aggregating *D. discoideum* cell densities are typically 250 per  $\text{mm}^2$  to  $1 \times 10^4$  per  $\text{mm}^2$  (Dallon and Othmer, 1997). Due to the computational resources required to manage a population of cytotobots within the upper range, two cell densities of 250 agents per  $\text{mm}^2$  (1250 agents) and 150 per  $\text{mm}^2$  (750 agents)

were chosen. The agents are initialized at random positions within the simulated environment. Each starving agent emits a cAMP signal at equal strength around its circumference into the environment. This signal is detected by other agents within or equal to  $r$ . In these experiments a range of  $r$  values are explored, including that of real cells of 1, 0.5, and 0.1 mm (McCann, et al., 2010). The actual cAMP signal degrades linearly with increasing distance ( $d$ ) from the emitting cell. Each agent detects the cAMP signal of all starving cells within or equal to  $r$ , and a total value for each direction ( $A$ ) is calculated using the pseudocode given in figure 3. Each cycle represents 1 minute of time. In this time the agent moves  $9\text{ }\mu\text{m}$ - a distance which corresponds to that reported in the literature (Rifkin and Goldberg, 2006). Therefore, after 1 hour motion the agent travels a distance of  $540\text{ }\mu\text{m}$ . In reality there are always remaining cells that do not aggregate, and thus the simulation runs until 95% of agents are at a distance of less than 0.1mm from their nearest neighbor.

## Results

The results for all 10 experiments are given in table 1. Each experiment was performed 100 times. In experiments 8, 9, and 10 the value of  $r$  and  $d$  are within the ranges reported for real *D. discoideum* cells. These experiments are used to compare the behaviors and length of time taken to aggregate with the literature. In experiments 8, and 9 aggregation completes after an average formation of 4.3 mounds in 10.05 hours, and 6.7 mounds in 12.65 hours respectively. In nature the organism takes between 9-13 hours to aggregate (Cotter, et al., 1992; Becker, et al., 2010), thus the results of these experiments have an aggregation time within the reported range. This is also comparable to other simulations. For example, Becker et al. (2010) reports an aggregation time of 11.6 hours for a simulated population of *D. discoideum* with a cell density of  $200\text{ mm}^2$ . In experiment 10, the population never satisfied the criteria for completion of aggregation, where instead the agents appeared to move in a fashion reminiscent of Brownian motion. The likely explanation for this is twofold. Firstly, the simulation does not consider the effect of glycoproteins where aggregating cells making contact with each other attach together. Secondly, because the attraction range is so small, agents are only able to detect other agents within their immediate neighborhood, thus momentarily larger clusters with higher attraction strength go undetected and quickly dissipate- an effect that would not occur if agents stayed together. The complete set of results shows that by increasing  $p$  by  $100\text{ mm}^2$  the number of mounds formed at each  $r$  decrease with the exception of experiment 6. This is not surprising, as denser populations should have more chance of interacting, and thus form fewer clusters, but with higher numbers of agents. Similarly, decreasing  $r$  results in a general increase to the number of mounds formed at both values of  $p$ . The likely reason for this is that as  $r$  decreases the agent becomes unable to influence increasing quantities of area, thus larger numbers of stable clusters can form but with fewer numbers of agents.

Emergent behaviors and clustering patterns similar to the biological organism were also observed. As previously discussed, the cytotobots are polarized.



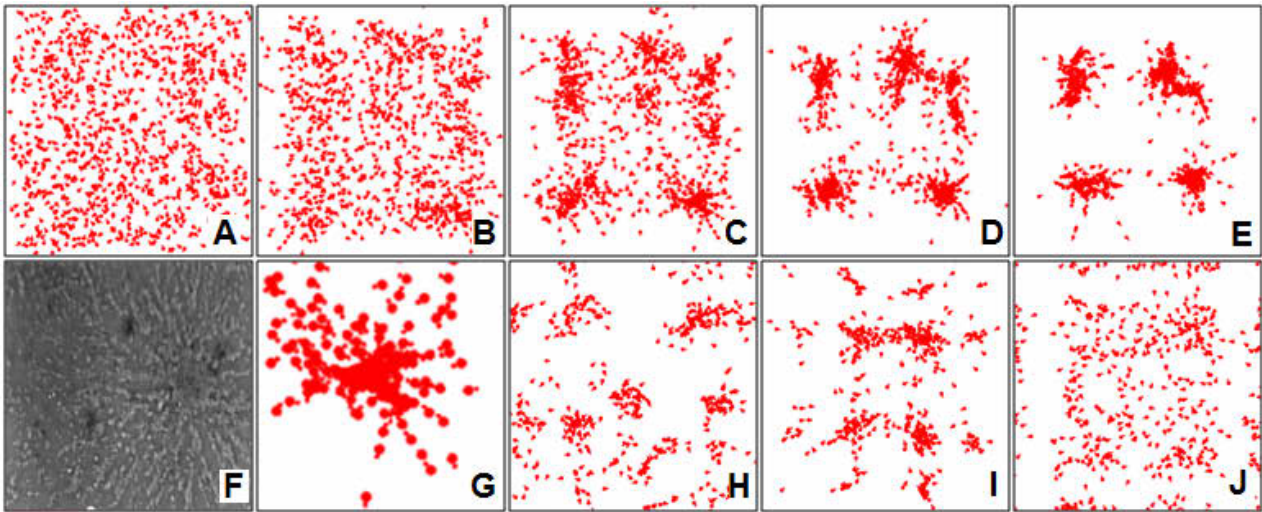


Figure 4: A-E Cytobot aggregation for experiment 8 at: A- 1hr, B- 2hr, C-5hrs, D- 8hrs, E- 12hrs; Image F- *D. discoideum* cells aggregating; G- Is the lower right hand corner of image C demonstrating streaming behavior; H- Spiral patterns in experiment 4 after 8 hours; I- Symmetrical patterns for experiment 2 at 7 hours; J- Wave pattern for experiment 2 at 2 hours.

Diagram F is courtesy of T. Gregor, Laboratory for the Physics of Life, Princeton University, 2013 Used with permission.

Table 1: Aggregation experiment simulation results

| No | Density (p) per mm <sup>2</sup> | Range (r) in mm | Mean No. of mounds; ( $\sigma$ = Sta. Dev.) | Mean time (hours); ( $\sigma$ ); *Literature Range 9-13 hours |
|----|---------------------------------|-----------------|---------------------------------------------|---------------------------------------------------------------|
| 1  | 150                             | 5               | 1<br>(0)                                    | 8.98<br>(0.09)                                                |
| 2  | 150                             | 2.5             | 4<br>(0.31)                                 | 9.63<br>(0.17)                                                |
| 3  | 150                             | 1               | 5.2<br>(0.82)                               | 9.92<br>(0.34)                                                |
| 4  | 150                             | 0.5             | 8.4<br>(1.19)                               | 10.23<br>(0.59)                                               |
| 5  | 150                             | 0.1             | 14.2<br>(2.36)                              | 10.6<br>(1.82)                                                |
| 6  | 250                             | 5               | 1<br>(0)                                    | 8.95<br>(0.11)                                                |
| 7  | 250                             | 2.5             | 1<br>(0)                                    | 9.6<br>(0.20)                                                 |
| 8  | 250                             | 1               | 4.3<br>(0.37)                               | 10.05<br>(0.58)                                               |
| 9  | 250                             | 0.5             | 6.7<br>(1.62)                               | 12.65<br>(1.94)                                               |
| 10 | 250                             | 0.1             | -                                           | -                                                             |

Implementing the agents in this way allowed us to observe whether or not the previously described streaming behavior occurs. A close up of the right hand corner of screenshot C is shown in figure 4G showing agents beginning to form a cluster. The protruding head of each agent can be seen clearly, where each lines up its head to the rear of another agent and forms a stream. As can be seen in figure 4F this is very similar to the streaming behavior in real cells of *D. discoideum*. Other emergent patterns occurred during different experiments including spirals (figure 4H), symmetric patterns (figure 4I), and waves (figure 4J).

### Oil Spill Clean-up Simulation

To illustrate a practical application, the cytotobots are used to tackle a simplified oil-spill clean-up simulation. In these experiments, the same ARN used previously produces different behaviors by altering its interface with the environment. In the following 4 experiments the length of time it takes for 3, 5, 8 and 15 cytotobots to clean up 95% of the oil is recorded. These results are compared with similar work.

The cytotobots move within a 2D simulated environment containing an oil spill. This oil is analogous to a distribution of food within a nutrient landscape. The task of the cytotobots is to clean up the spill as quickly as possible by consuming oil at each location. The agents move through the environment by switching between the two previously described behavioral modes- foraging and starvation. In the aggregation experiments, no food was present, thus the foraging behavior remained inactive. In this case, the concentration of oil surrounding the agents was fed into both the receptor pools of the weighted direction network and the food network. Thus in this case oil represents both food and cAMP. At the start of each experiment, the cytotobots are distributed randomly within the environment, and the ARN network is initialized as previously described. The agents start the simulation in foraging mode but during the simulation alternate between foraging and starvation modes. Starvation behavior is triggered after the most recent positions (minimum of 2) contained zero food. In starvation mode, instead of turning in a random direction, the new direction is weighted toward higher concentrations of food within its surrounding area. This behavior forces exploration of unexplored search space because previously visited positions have a food level of 0. Consumption of environmental food therefore acts as a stigmergic signal, where agents are inclined to move up the nutrient gradient created by their foraging activities. Here, we



model the spillage of 100 tonnes of Statfjord crude oil at  $15^{\circ}\text{C}$  under a wind speed of  $5\text{ms}^{-1}$ . The oil is distributed over a 2D sea surface of  $300\text{m}$  by  $200\text{m}$ , thus an area of  $60000\text{m}^2$ , where 2 pixels corresponds to  $1\text{m}$ , as shown in figure 4A. This particular oil type and parameter set were chosen in order to compare directly with work by Kakalis and Ventikos (2008) who present a robotic swarm concept for oil spill confrontation. For this reason, we account for an initial response time of 14 hours. Based on the complex mathematical models found in Kakalis and Ventikos which account for the main factors of short term changes in oil characterization, the volume of oil after 14 hours is reduced to  $150\text{m}^3$ . Beyond the starting state, the volume is only influenced by the cytobots. The speed of each agent is  $0.5\text{ms}^{-1}$  and is based on other robotic agents in oil cleaning scenarios (Kakalis and Ventikos, 2008), thus the cytobots move 1 pixel ( $0.5\text{m}$ ) for every time step. The actual cleaning surface is  $1\text{m}$ , thus the cytobots clean a 2 pixel wide area in each time step.

Mathematical modeling of an oil spill is non-trivial and at best can offer a crude approximation of its actual trajectory. Most oil spills quickly form a comet shape with most of the oil within the head and a trail of sheen (Wang and Stout, 2007). To represent a simplified version of the comet shaped spread, the area is divided into  $100\text{m} \times 200\text{m}$  segments. The first segment contains  $0.015$  tonnes of oil, and each subsequent segment increases by  $0.03$  tonnes from right to left.

## Results

In each experiment, a different number of cytobots was deployed- 3, 5, 8 and 15 and the recovery rate achieved by each group was compared. The simulation time was measured from deployment of the cytobots at 0 hours (14 hours after oil was spilled) and stopped when the cytobots had collectively removed 95% of the  $150\text{m}^3$  of oil. Each experiment was run 100 times, and the average volume of oil consumed at 6 minute intervals was calculated. Figure 6 presents the volume of oil consumed by each group of cytobots against time. The finishing times in hours are 15.2, 11.5, 9.6, and 6.1 for 3, 5, 8, and 15 cytobots respectively. By adding 2 additional agents to the group of 3 the length of time is reduced by 3.5 hours, thus  $1.75$  hour difference per extra cytobot. This difference decreases  $1.12$  hours per cytobot for 8 agents, then to  $0.76$  per agent for 15. This variation can be accounted for by examining the agents' paths through the oil. Rates are much faster at the beginning of the experiments, where cytobots move toward the oil rich left side of the environment. This can be seen in the series of screenshots in figure 5 where A shows the starting position at time 0, and B shows that after 2 hours the cytobots have moved toward the left-hand side, focusing mainly on highly concentrated areas (consumed oil is shown in white). Initially, the rate of oil removal is high because cytobots focus on the volume rich areas and cannot go over their path, thus each new location results in consumption of oil. However, as time progresses, large patches become cleaned and a higher probability exists for the cytobots to revisit previously cleaned areas. The consumption of oil in figure 5 C and D at 4 and 9.6 hours respectively shows more clearly that cytobots focus cleaning efforts on the richest volume area first, and are gradually forced to move

toward the next highest concentration by the gradient created by their foraging activities.

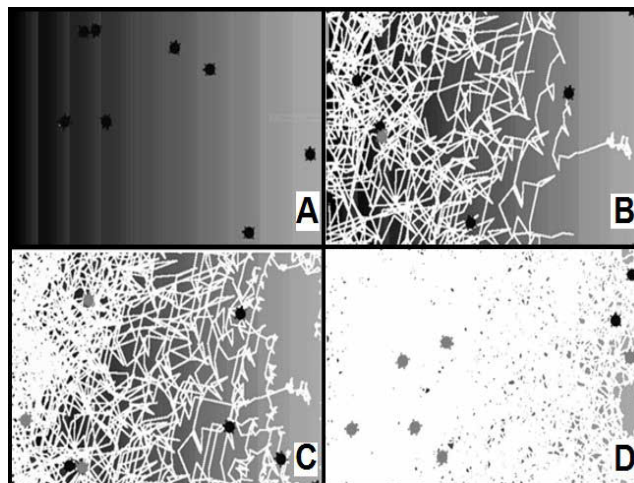


Figure 5: Oil simulation using 8 cytobots at A- 0 hours, B- 2 hours, C- 4 hours and D- 9.6 hours

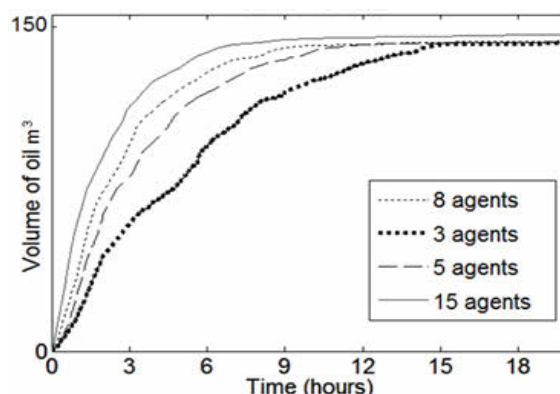


Figure 6: Volume of oil cleaned against time for each group of cytobots

Figure 5 D shows the state of the oil at the end of the simulation, where only small patches remain mainly in areas of low oil volumes. These results can be compared to the simulation by Kakalis and Ventikos. Here, varying numbers of simulated EU-MOP robots are deployed to tackle  $150\text{m}^3$  of Statfjord oil over  $60000\text{m}^2$  (as before). In this case, the robots have a slightly faster speed of  $0.54\text{m/s}$  but have the same  $1\text{m}$  skimming face. Each EU-MOP robot has a storage capacity of  $2\text{m}^3$  and a transit speed of  $2.1\text{ms}^{-1}$ . The time taken for 3, 5, 8, and 15 EU-MOPS are 54, 32, 20 and 10 hours respectively. For comparison, the results of our simulation can be adjusted to include unloading of the oil at a servicing vessel. Using the same storage capacity and transit speed and assuming the distance to the ship and back is 2 times  $300\text{m}$  and that each cytobot fills the same amount simultaneously, then the new times are 17.2, 12.7, 10.3 and 6.5 for 3, 5, 8 and 15 cytobots respectively. The Kakalis and Ventikos simulation has several differences to the one reported here, particularly in the distribution of the oil. Also, some key parameters are missing from their paper, for example, distance to boat. Despite these

differences, our results are very similar. For example the reported simulation time for 15 EU-MOPS is 10 hours and in our simulation 5 and 8 cybotbots took 12.7 and 10.3 hours respectively. Given the differences in the simulation and differences in operation of the robots, the resulting clean up times are comparable showing that the cybotbots have potential application as distributed robotic agents in real-world environments.

## Conclusions

The aggregation experiment results presented above show that the agents are able to simulate behavior of individual unicellular organisms, and model emergent behavior arising from interactions among such groups. These results demonstrate the parallels between ARN agents and the biological counterpart from which they were inspired. It also highlights a potential use as a means to simulate groups of interacting cells such as a bacterial colony or tissue component within a multicellular organism.

The results for the oil spill simulation demonstrate potential application for the ARN agents as autonomous agents within real world environments. This application demands an internal control system which can function without reference to other agents within the environment which are operating in parallel. By modifying the environment, (which in this case was consumption of food), the agents can stigmergically communicate and facilitate emergent behavior. The cybotbots offer a unique range of abilities. Like cells, their internal network of spatially distributed dynamic chemical species allows them to autonomously coordinate and direct their movement, recognize and respond to patterns in the environment, and produce high-level behavior.

In future work, it is intended to further explore the AI applications of the cybotbot agents, and later, to create swarms of cybotbot robots with applications in real world environments.

## References

- Arkin, A. and Ross, J. (1994) Computational functions in biochemical reaction networks, *Biophysical Journal*, 67:560-578.
- Becker, M. (2010). Simulation model for the whole life cycle of slime mould *Dictyostelium Discoideum*. In proceedings of the European conference on modeling and simulation, pages 247-253.
- Bhalla, U. S. (2003). Understanding complex signaling networks through models and metaphors. *Progress in Biophysics and Molecular Biology*, 81(1):45-46.
- Bray, D. (1995). Protein molecules as computational elements in living cells. *Nature*, 376(6538):307-12.
- Cotter, D. A., Sands, T. W., Virdy, K. J., North, M. J., Klein, G., Satre, M. (1992). Patterning of development in *Dictyostelium discoideum*: factors regulating growth, differentiation, spore dormancy and germination. *Biochemistry and Cell Biology*, 70(10-11):892-919.
- Dallon, J. C. and Othmer, H. G. (1997). A discrete cell model with adaptive signaling for aggregation of *Dictyostelium discoideum*. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 352(1351): 391-417.
- Dittrich, P., Zeigler, J., Banzhaf, W. (2001). Artificial chemistries- a review. *Artificial Life*, 7(3):225-275.
- Ford, B. J. (2009). On intelligence in cells: The case for whole cell biology. *Interdisciplinary Science Reviews*, 34(4):350-365.
- Gerrard, C. E., McCall, J., Coghill, G. M. and Macleod, C. (2011). Artificial Reaction Networks, In Proceedings of the 11<sup>th</sup> UK Workshop on Computational Intelligence, pages 20-26.
- Gerrard, C. E., McCall, J., Coghill, G. M. and Macleod, C. (2012a). Adaptive dynamic control of quadrupedal robotic gaits with Artificial Reaction Networks. In Proceedings of the 19<sup>th</sup> international conference on neural information processing, pages 280-287.
- Gerrard, C. E., McCall, J., Coghill, G. M. and Macleod, C. (2012b). Temporal patterns in artificial reaction networks, In Proceedings of the 22nd international conference on Artificial Neural Networks, pages 1-8.
- Kakalis, N. M. P., Ventikos, Y. (2008). Robotic swarm concept for efficient oil spill confrontation. *Journal of Hazardous Materials*, 154(1-3):880-7.
- McCann, C. P., Kriebel, P. W., Parent, C. A., Losert, W. (2010). Cell speed, persistence and information transmission during signal relay and collective migration. *Journal of Cell Science*, 123:1724-1731.
- Patidar, V., Sud, K. K., and Pareek, N. K. (2009). A pseudo random generator based on chaotic logistic map and its statistical testing. *Informatica*, 33:441-452.
- Rifkin, J. L. and Goldberg, R. R. (2006). Effects of chemoattractant pteridines upon speed of *D. discoideum* vegetative amoeba. *Cell Motility and the Cytoskeleton*, 63(1): 1-5.
- Tyson, J. J. and Novák, B. (2010). Functional motifs in biochemical reaction networks. *Annual Review of Physical Chemistry*, 61:219-240.
- Ulam, S. M. and von Neumann, J. (1947). On combinations of stochastic and deterministic processes. *Bulletin of the American Mathematical Society*, 53:1120.
- Vladimirov, N. and Sourjik, V. (2009). Chemotaxis: how bacteria use memory. *The Journal of Biological Chemistry*, 390(11):1097-1104.
- Wang, Z. and Stout, S. A. (2007). Oil spill environmental forensics: fingerprinting and source identification. Academic Press, Burlington MA.

# Exploiting Dynamical Complexity in a Physical Tensegrity Robot to Achieve Locomotion

Mark Khazanov<sup>1</sup>, Ben Humphreys<sup>1</sup>, Willam Keat<sup>2</sup> and John Rieffel<sup>1</sup>

<sup>1</sup>Computer Science Department, Union College, Schenectady NY 12308

<sup>2</sup>Mechanical Engineering Department, Union College, Schenectady NY 12308  
rieffelj@union.edu

## Abstract

The emerging field of *morphological computation* seeks to understand how mechanical complexity in living systems can be advantageous, for instance by reducing the cost of control. In this paper we explore the phenomenon of morphological computation in tensegrities – unique structures with a high strength to weight ratio, resilience, and an ability to change shape. These features have great value as a robotics platform, but also make tensegrities difficult to control via conventional techniques. We describe a novel approach to the control of tensegrity robots which, rather than suppressing complex dynamics, *exploits* them in order to achieve locomotion. Our robots are physically embodied (rather than simulated), evolvable, and locomote at higher speeds (relative to body size) and with fewer actuators than those controlled by more conventional approaches.

## Introduction

Traditional engineering approaches to design of structures and the control of robots attempt to avoid, or at least actively suppress, complex system dynamics such as vibration and dynamic coupling among components. By reverting them to generally rigid and holonomic systems, their kinematics can be modeled using classical mechanics (Murray et al., 1994; Sciavicco and Siciliano, 2000), and these models can in turn be used to generate desired movements.

By contrast, the bodies of many natural organisms (and robots which attempt to mimic them) are by their very nature high dimensional dynamic systems with an essentially infinite number of degrees of freedom. Properties of living systems such as elasticity and deformability come at the cost of resonances and tight dynamic coupling between components (Trimmer, 2007) – properties which are often assiduously avoided in conventional engineering approaches to robotic design. This precludes the use of most of the traditional kinematic and inverse-dynamics approaches described above (Craig, 1989). While some methods exist for the control of non-holonomic (under-controlled) mechanical systems, many are incredibly computationally expensive, and difficult to transfer from simulation to reality (Hannan and Walker, 2003; Fung, 1993; Vogel, 2003).

How then, are dynamically complex biological systems so controllably robust and agile? The emerging field of *morphological computation* (Paul, 2006; Pfeifer et al., 2007; Pfeifer and Bongard, 2006) conjectures that “outsourcing” the computation into the mechanics of the structure allows related neural pathways to devote their resources to higher level tasks (Valero-Cuevas et al., 2007) – a type of “intelligence by mechanics” (Blickhan et al., 2007). These phenomena have been shown in the physiology of animals such as wallabies (Biewener et al., 2004) and guinea fowl (Daley and Biewener, 2006) and cockroaches (Ahn and Full, 2002).

Biological morphological computation has served as inspiration for robotic control in several recent works. Iida and Pfeifer (Iida and Pfeifer, 2006) explored how the body dynamics of a quadruped robot can be exploited for sensing. Watanabe (Watanabe et al., 2008) demonstrated how inducing long distance mechanical coupling in a snake robot improves its ability to learning a crawling motion. All of these systems were largely composed of rigid elements. Our interest is expanding these principles into the realm of soft materials and structures, where the complexity, and therefore the potential for beneficial exploitation, is significantly higher.

This paper demonstrates morphological computation in tensegrity robots. Tensegrities, pre-stress stable structures composed of rigid struts and tensile springs, possess many appealing traits, but exhibit high degrees of mechanical coupling, and are therefore difficult to control through conventional means. We implement an alternative and novel approach to tensegrity locomotion, one which seeks to exploit, rather than suppress, vibration and dynamical coupling between components. The resulting robot is quite simple, with open-loop actuation by low-voltage vibrating pager motors, and yet is capable of robust and controllable motion. We believe this to be both the fastest and the smallest physically embodied tensegrity robot yet to be developed. We begin the paper by describing the design of the robot. We then demonstrate vibrationally-actuated gaits which produce linear and rotational behavior. These gaits can be sequenced together in order to generate controllable trajectories. We conclude by discussing more sophisticated control algorithms and op-



timization techniques.

## Tensegrity Robots

A tensegrity structure (Figure 1) is a self-supporting structure consisting of a set of disjoint rigid elements (struts) whose endpoints are connected by a set of continuous tensile elements (springs). Despite the fact that none of the rigid elements touch, tensegrities are able to maintain their structure due to a synergistic interplay of compressive and tensile forces (Wang, 1998). Because of this *pre-stress stability*, they are able to quickly return to form when perturbed by an outside force. (Connelly and Back, 1998).

Examples of tensegrity can be found in structures ranging from camping tents to sports stadiums. These same principles are also found in the biological realm, at all scales from the structure of proteins (Ingber, 1998) and cellular cytoskeleton (Wang et al., 2001) up to the tendinous network of the human hand (Valero-Cuevas et al., 2007).

What makes tensegrities particularly appealing as a robotic platform is their high strength-to-weight ratio and resilience, as well as their ability to change shape by altering the resting length of the tensile elements. As a result, tensegrity structures are increasingly being used for applications such as smart structures and soft robots (Tibert and Pellegrino, 2003; Tibert, 2002; Motro, 2003; Sultan, 1999; Matsuda and Murata, 2006).

Unfortunately, the pre-stress stability of tensegrities imposes complex nonlinear dynamics, even for relatively small structures (Skelton et al., 2001). Conventional approaches to tensegrity robots therefore attempt to dampen the vibrational modes of the robots before controlling them. Skelton *et al.* have been able to demonstrate both active vibration damping (2004) and open-loop control of simple structures (2004).

In most cases, once the vibration and dynamical coupling of a tensegrity robot has been reduced either actively or passively, deformation and control are achieved by changing the rest lengths of the tensile elements, for instance by attaching strings to a reeled servo motor (Paul et al., 2006, 2005). Even so, the majority of tensegrity robotics has occurred in simulation (Aldrich et al., 2003; Paul et al., 2006; Graells Rovira and Mirats Tur, 2009; Iscen et al., 2013) rather than reality (Shibata et al., 2009). One notable recent contribution is Caluwaerts *et al.*'s work (2013) on physical reservoir computing in a simulated tensegrity robot, in which they demonstrate learnable gaits produced by relatively simple central pattern generators (CPGs).

There are a few published examples of physically embodied tensegrity robots moving. Paul *et al.* (2006) built a three-bar tensegrity robot with 0.4m struts, with three of its tensile elements actuated by servomotors. Using a gait derived from simulation, the physical robot was able to achieve speeds of around 60cm/min. Shibata *et al.* (2009) built a 6-bar tensegrity with 15cm struts, with its tensile elements actuated by shape-memory alloy wires. (The speed of the

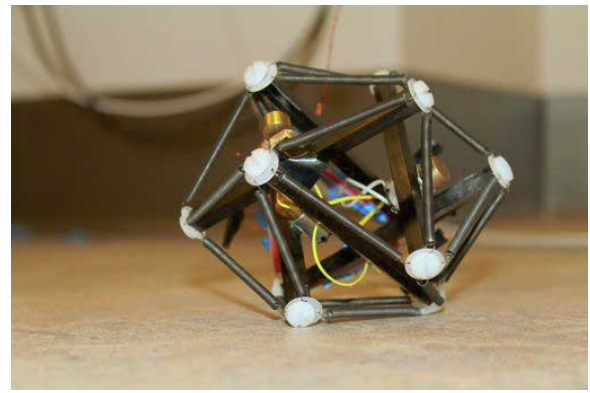


Figure 1: Tensegrities consist of a set of rigid elements (rods) joined together by tensile elements (springs). They maintain their shape through a synergistic interplay of forces. (Photo by Steven Stangle)

resulting robot was not published). Later, Koizumi *et al.* (2012) built a six-bar robot with 0.6m struts with tensile elements actuated by 24 pneumatic McKibben actuators, which moved by rolling. While the speed of the robot is not published, an online video shows 15 rolls in the space of 45 seconds (1 roll per 3 seconds). We estimate from the video that the robot is moving at approximately 25cm per roll, or 5m/min.

## A new way to move tensegrities

The approaches describe above have essentially treated tensegrity-based robots as quasi-static and non-oscillating structures. And yet, tensegrities are by their very nature highly dynamic – anecdotally, the tensegrities we have built in our lab will readily oscillate as the table is bumped, or even as someone types on a keyboard.

The motivation for our work, therefore, lies in striving to exploit, rather than suppress, this inherent dynamical complexity as an advantage – making tensegrities move by vibrating, rather than suppressing their vibrations. Since the dynamics of real-world tensegrity structures are incredibly difficult to model in simulation, we chose to avoid simulation entirely and perform all experiments in the physical world. Quoting Rodney Brooks, “the world is its own best model” (Brooks, 1990).

## Design

Our ambition was to design a small (< 15 cm) tensegrity that was powered by vibration alone. It would also have to be robust enough to endure the long hours of testing and, given the practical challenges of constructing a mechanical device on so small a scale, it would be advantageous if it were easy to manufacture and repair.

The resulting design, based upon a canonical six-bar tensegrity shape, is shown in Figure 1. The geometry is



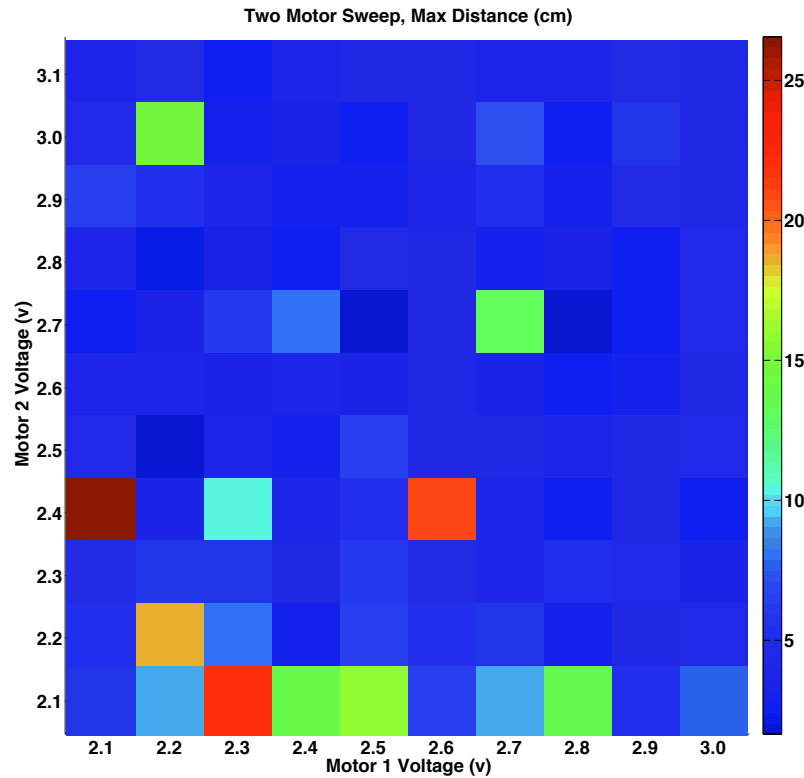


Figure 4: Max distance (over 10 evaluations, each lasting 7 seconds) traveled by the robot during a sparse sweep of voltages for motors 1 and 2 between 2.1V and 2.9V, in increments of 0.3V (motor 3 was fixed at 0V). As can be seen, the resulting space is complex, and distance does not correlate directly with motor speed.

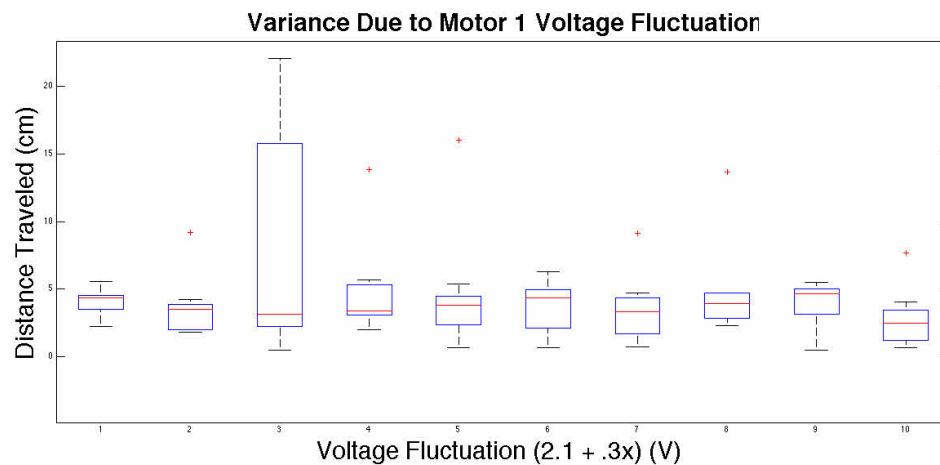


Figure 5: Variation across trials for a sweep of motor 1 values while keeping voltages for motors 2 and 3 fixed (the bottom row of Figure 4))



Figure 6: An example of how distance traveled was automatically measured. Position was calculated by subtracting an “empty” arena image from one containing the robot. Distance could then be measured by comparing beginning (left) and final (center) robot positions (right).



Figure 2: The robot testing arena is on the floor in order to maximize the navigable area. (Photo by Steven Stangle)



Figure 3: The body of the tensegrity robot consists of rigid plastic rods and metal springs. It is actuated by simple DC vibrating motors attached at the midpoint of three of its six struts. A closeup of the robot illustrated the springs and the DC motors. (Photo by Steven Stangle)

defined by six equal length composite struts which are connected to each other via 24 identical helical springs, with four springs emanating from each strut end.

This 6-bar tensegrity has three orthogonal planes of symmetry. As a result of this natural symmetry, each spring wants to stretch an equal amount in the fully assembled equilibrium configuration, greatly simplifying design and analysis. There is however a slight loss of symmetry when the tensegrity is placed in contact with the ground because none of the stable contact positions are aligned with the planes of symmetry. Sagging of the springs under the weight of the spars and motors further disrupts the symmetry. Three small vibration motors were mounted on composite struts and oriented such that each shaft axis is perpendicular to one of the planes of symmetry.

Few actual machining operations are required to produce the tensegrity. The 9.4 cm long composite struts are cut from 6.35 mm square graphite composite tubes. The motors were mounted to the flat outer surface of the struts using epoxy. The struts, while square on the outside, are hollow circular on the inside. Therefore, both ends of each strut could be tapped to allow for insertion of 10-24 nylon screws. These screws provide a smooth contact surface for the tensegrity to rest on and are used to fasten nylon washers to the ends of the struts. The hooked ends of the helical springs are attached directly to the nylon washers via 4 equally-spaced drilled holes.

Selection of the springs is critical to overall performance. The basic strategy was to try to produce the smallest possible natural frequencies under the presumption that small natural frequencies would lead to large displacement amplitudes, thus enhancing the chances that the tensegrity might roll. We strove to achieve this goal by minimizing spring stiffness and spring preload subject to the constraint of keeping static deflections within an acceptable limit (so that the tensegrity would not lose its basic shape). A single vertical strut was modeled as supported by eight linear springs oriented at  $45^\circ$  in order to limit the maximum static deflection

to 5% of total strut length. This calculation led to selection of a helical spring with a spring constant of 0.209 N/cm.

Attempts were also made to optimize the selection of a vibration motor. While the need for a small motor with a relatively large offset mass was clear, choosing the best range of operating speeds involved compromise. On the one hand, the motor speeds have to be large enough to produce sufficient centrifugal force, but also small enough to excite the lower (high energy) natural frequencies of the tensegrity. These considerations led to selection of the Precision Microdrives vibration motor Model 312-107 (Figure 3) which operates between 100 and 260 Hz.

To check the suitability of this operating range, the tensegrity was modeled using a matrix structural analysis code. One end of one strut was assumed fixed in space and the associated natural frequencies were determined. The fundamental (lowest) frequency was found to be 7.8 Hz, well below the operating range of the motor. However, physical testing of the tensegrity with these motors indicated the existence of natural frequencies that were within the operating range.

## Evaluation

Having designed the robot in order to maximize resonance, we then evaluated its ability to locomote via vibration. Early trials indicated a wide diversity of motions was possible, and that small changes in motor frequencies could lead to large changes to the resulting gait. Below we seek to quantify this.

(Videos of the robot moving can be seen at the corresponding author's web page: [www.cs.union.edu/~rieffelj/videos/](http://www.cs.union.edu/~rieffelj/videos/))

## Setup

The robot was placed into an 85cm x 80cm arena (Figure 2) on the floor of our lab (the robot had a habit of falling off of tables) and tethered to the power supply and motor controllers. In order to reduce the effects of the tether, which might undesirably constrain the motion, the tether was built from narrow gauge magnet wire. Two cameras, a USB camera and a small handheld video camera, were placed 130cm above the arena. The USB camera was used for distance measures and the video camera was used to document results.

The voltage (speed) of each of the three motors was controlled by USB motor controllers connected to a host computer.

As illustrated by Figure 6, the process of distance measure was automated using the overhead USB camera connected to the control computer. The location of the tensegrity in a frame was determined by subtracting the image of an "empty" arena from an image containing the robot and then finding the centroid of the remaining pixels. The tether is visible in some frames, but has a negligible impact upon positional measurements. Distance could then be calculated

by comparing the pre- and post-evaluation locations. The arena was large enough that multiple evaluations could often be performed before manually returning the robot to the center of the arena.

## Two-Motor Gaits

In order to demonstrate the diversity of gaits produced by the tensegrity robot, we ran a sparse sweep of motor voltages for two motors between 2.1V and 2.9V, keeping the third motor fixed at 0V (voltages below 2.0V do not produce motion in the motors). Each frequency set was evaluated over ten trials, each lasting 7 seconds. Figure 4 illustrates the maximum distance traveled by the robot at each measured frequency pair.

Figure 5 shows the variation in distances between trials when sweeping through frequencies for motor 1 while keeping the other two motor frequencies fixed at 2.1V and 0V respectively (corresponding to the the bottom row of the heat map).

Combined, these results hint at the complexity of the underlying space of distances achievable by the full range of three motor voltages: even when only using two of the three motors, there is significant variation in distances traveled, and there is a non-linear relationship between motor frequencies and distance traveled. Lacking any analytical approach to mapping motor frequencies to corresponding gaits, this suggests that automated trial and error via hill climbing or genetic algorithm might be the best way to discover effective gaits.

## Three-Motor Gaits

Having demonstrated the diversity and complexity of two-motor gaits, we then manually explored three-motor gaits, searching for frequency sets which led to interesting and effective behaviors. We were able to discover sets which produced consistent clockwise, anti-clockwise, and linear locomotion.

Figure 7 illustrates the motion of the tensegrity over the course of 6 seconds of motion, at two-second intervals.

Maximum linear locomotion speed was on the order of 4cm/sec, or 2.4m/min. On an absolute basis this is four times faster than Paul's robot and half the speed of Kuizumi's. When normalized to body size, however, our tensegrity is considerably faster, while using simpler means of locomotion.

Videos of these gaits are available at the corresponding author's web page.

## Controllable Motion

Most compellingly, these frequency sets which result in diverse gaits can be sequenced in order to *steer* the tensegrity in a controllable fashion. Since one gait can be used to propel the robot forward, and a second to rotate it, these gaits can be sequenced in order to produce controllable motion.



Figure 7: Video frames, taken at two-second intervals, illustrating motion of the vibrating tensegrity. Given a single set of motor frequencies, the robot is able to exploit its vibrational tendencies in order to move quite quickly – at speeds over 4cm/sec. Videos available on the author’s web site.

To demonstrate this we created an alternating sequence of forward-propagating and rotational gaits which caused the tensegrity robot to traverse a path within the arena. A video of this trajectory is provided on our web page.

This method of steering a tensegrity robot simply by changing its mode of vibration is unique, and a valuable example of how adding morphological complexity can sometimes simplify the task of control, allowing aspects of control to be “outsourced” into the complex mechanics of a structure. This is, therefore, a valuable example of *morphological computation* in a tensegrity robot.

As we continue our studies, and scale to increasingly large and complex tensegrity robots, we hope to uncover an even more diverse range of gait behaviors, leading to more interesting and effective means of controlled locomotion.

## Discussion

There are several improvements we would like to make to move the system forward. Foremost among them is automating the discovery of effective motor frequencies with a physically embodied evolutionary algorithm, following in the footsteps of Harvey *et al.* at Sussex (Harvey *et al.*, 1997), Watson *et al.* (Watson *et al.*, 1999), and more recently Zykov (2004) and Yosinski (2011).

Like all physically embodied evolutionary robotics, however, we must deal with the issue of noisy evaluation, reliability, and consistency between trials. Some solutions lie at the algorithmic level (for instance, via multiple trials (Fitzpatrick and Grefenstette, 1988)), and some lie at the hardware level, for instance by using a more consistently smooth surface for evaluation. However, we want to avoid “sterile” surfaces – perfectly flat, perfectly smooth – since our ambition is to evolve robots capable of robust performance in rough and uncertain environments.

There is also the matter of the evolvability of the system itself. While the dynamics of tensegrities are complex enough to justify the use of automated optimization techniques (as opposed to analytical modeling) to generate gaits, our current control scheme is not very amenable to evolution. The use of a single control parameter (frequency) means that genotypes contain only three floating

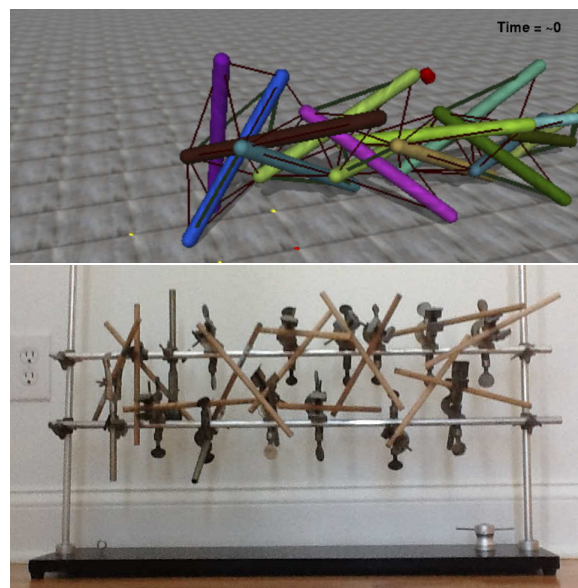


Figure 8: *top*: A model of a larger 15-bar tensegrity, from (Rieffel *et al.*, 2010). *bottom*: a rig we have built to enable construction of the physical 15-bar tensegrity.

point loci, making the system somewhat too simple to explore with GAs. In the near term we are interested in more complex open-loop gaits, for instance, allowing each motor’s control voltage to change during a gait, and being able to specify its phase, amplitude, and frequency. In the longer term, we are interested in closed-loop control via Artificial Neural Networks (ANNs), with feedback provided by off-the-shelf micro-scale accelerometers (such as those used in smart phones).

We are also interested in using high speed video to analyze in more detail the varying resonant modes exhibited during different gaits.

Ultimately our goal is to build tensegrity robots with considerably more structural elements (while maintaining our relatively short strut length), such as the one shown in Figure 8. As tensegrities become more complex and irregular, they become increasingly difficult to analyze or model, fur-



ther necessitating our embodied approach.

## Conclusion

Tensegrities are an appealing platform for modern robotics. They are robust, agile, and can quickly change shape, lending themselves to promising applications ranging from urban search-and-rescue to biomedical devices. However, these properties also make them exceedingly difficult to control through conventional means, particularly as the complexity of the robot increases. We have described a means of actuating and controlling tensegrity robots which treats their dynamical complexity as a feature to be exploited rather than as a liability to be suppressed. By designing the structure in order to maximize resonant possibilities, we can make the robot move simply by vibrating it at specific frequencies. This leads to a tensegrity robot which is much smaller and much simpler than existing designs, and yet outperforms in many regards.

More valuably, we have demonstrated how we can affect behavioral change merely by changing the frequencies at which our robot vibrates. Achieving behavioral diversity by exploiting mechanical complexity in this manner is a valuable example of *morphological computation*, in which increasing dynamical coupling can, paradoxically, reduce the cost of control. Given the pervasiveness of both tensegrity and dynamical coupling in biological systems, our hope is that this can lead to a deeper understanding of how mechanically complex living systems at all scales of life move and interact with the world.

## Acknowledgments

The authors would like to thank RJ Hojnacki, Steve Stangle, and Julian Jocque, all of whom assisted in this research as part of undergraduate practica at Union College. Funding for undergraduate summer research by authors Mark Khazanov and Ben Humphreys was provided through the generosity of the Union College Summer Research Fellowship Program. Materials were paid for in part by several Union College Internal Education Fund (IEF) grants to the authors.

## References

- Ahn, A. N. and Full, R. (2002). A motor and a brake: two leg extensor muscles acting at the same joint manage energy differently in a running insect. *Journal of Experimental Biology*, 205.
- Aldrich, J., Skelton, R., and Kreutz-Delgado, K. (2003). Control synthesis for a class of light and agile robotic tensegrity structures. In *American Control Conference, 2003. Proceedings of the 2003*, volume 6, pages 5245–5251. IEEE.
- Biewener, A., McGowan, C., Card, G., and Baudinette, R. (2004). Dynamics of leg muscle function in tammar wallabies (*Macrotis eugenii*) during level versus incline hopping. *Journal of Experimental Biology*, 207.
- Blickhan, R., Seyfarth, A., Geyer, H., Grimmer, S., and Wagner, H. (2007). Intelligence by mechanics. *Phil Trans R. Soc. A*, pages 199–220.
- Brooks, R. A. (1990). Elephants don't play chess. *Robotics and Autonomous Systems*, 6:3–15.
- Caluwaerts, K., D'Haene, M., Verstraeten, D., and Schrauwen, B. (2013). Locomotion without a brain: Physical reservoir computing in tensegrity structures. *Artificial life*, 19(1):35–66.
- Chan, W. L., Arbelaez, D., Bossens, F., and Skelton, R. E. (2004). Active vibration control of a three-stage tensegrity structure. In *Proceedings of SPIE 11th Annual International Symposium on Smart Structures and Materials*.
- Connelly, R. and Back, A. (1998). Mathematics and tensegrity. *American Scientist*, 86.
- Craig, J. (1989). *Introduction to Robotics*. Addison-Wesley.
- Daley, M. and Biewener, A. (2006). Running over rough terrain reveals limb control for intrinsic stability. *Proceedings of the National Academy of Sciences*, 103:15681–15686.
- Fitzpatrick, J. M. and Grefenstette, J. J. (1988). Genetic algorithms in noisy environments. *Machine learning*, 3(2-3):101–120.
- Fung, Y. (1993). *Biomechanics: mechanical properties of living tissues*. Springer, 2 edition.
- Graells Rovira, A. and Mirats Tur, J. M. (2009). Control and simulation of a tensegrity-based mobile robot. *Robotics and Autonomous Systems*, 57(5):526–535.
- Hannan, M. and Walker, I. (2003). Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots. *J Robot Syst*, 20:45–63.
- Harvey, I., Husbands, P., Cliff, D., Thompson, A., and Jakobi, N. (1997). Evolutionary robotics: the sussex approach. *Robotics and autonomous systems*, 20(2):205–224.
- Iida, F. and Pfeifer, R. (2006). Sensing through body dynamics. *Robotics and Autonomous Systems*.
- Ingber, D. E. (1998). The architecture of life. *Scientific American*.
- Iscen, A., Agogino, A., SunSpiral, V., and Tumer, K. (2013). Controlling tensegrity robots through evolution. In *Proceedings of the 2013 Conference on Genetic and Evolutionary Computation (GECCO '13)*.
- Koizumi, Y., Shibata, M., and Hirai, S. (2012). Rolling tensegrity driven by pneumatic soft actuators. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1988–1993. IEEE.
- Masic, M. and Skelton, R. E. (2004). Open-loop control of class-2 tensegrity towers. In *Proceedings of SPIE 11th Annual International Symposium on Smart Structures and Materials*.
- Matsuda, T. and Murata, S. (2006). Stiffness distribution control-locomotion of closed link robot with mechanical softness. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1491–1498. IEEE.

- Motro, R. (2003). *Tensegrity: Structural Systems for the Future*. Kogan.
- Murray, R., Li, Z., and Sastry, S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.
- Paul, C. (2006). Morphological computation: A basis for the analysis of morphology and control requirements. *Robotics and Autonomous Systems*, 54.
- Paul, C., Valero-Cuevas, F. J., and Lipson, H. (2005). Redundancy in the control of robots with highly coupled mechanical structures. In *Int. Conf. on Intelligent Robots and Systems*, pages 802–808.
- Paul, C., Valero-Cuevas, F. J., and Lipson, H. (2006). Design and control of tensegrity robots for locomotion. *IEEE Transactions on Robotics*, 22(5).
- Pfeifer, R. and Bongard, J. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press.
- Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science*, pages 1088–1093.
- Rieffel, J. A., Valero-Cuevas, F. J., and Lipson, H. (2010). Morphological communication: exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. *Journal of The Royal Society Interface*, 7(45):613–621.
- Sciavicco, L. and Siciliano, B. (2000). *Modelling and Control of Robot Manipulators*. Springer.
- Shibata, M., Saijyo, F., and Hirai, S. (2009). Crawling by body deformation of tensegrity structure robots. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4375–4380. IEEE.
- Skelton, R. E., Pinaud, J. P., and Mingori, D. (2001). Dynamics of the shell class of tensegrity structures. *Journal of the Franklin Institute*.
- Sultan, C. (1999). *Modeling, design, and control of tensegrity structures with applications*. PhD thesis, Purdue University.
- Tibert, A. (2002). *Deployable Tensegrity Structures for Space Applications*. PhD thesis, Department of Mechanics, Royal Institute of Technology (KTH).
- Tibert, A. and Pellegrino, S. (2003). Review of form-finding methods for tensegrity structures. *International Journal of Space Structures*, 18(4):209–223.
- Trimmer, B. (2007). New challenges in biorobotics: incorporating soft tissue into control systems. In *IEEE International Conference on Robotics and Automation*.
- Valero-Cuevas, F., Yi, J., Brown, D., McNamara, R., Paul, C., and Lipson, H. (2007). The tendon network of the fingers performs anatomical computation at a macroscopic scale. *IEEE Trans Biomed Eng.*, 54:1161–6.
- Vogel, S. (2003). *Comparative biomechanics : life's physical world*. Princeton University Press.
- Wang, B.-B. (1998). Cable-strut systems: Part i - tensegrity. *J. Construct. Steel Res.*, 45(3).
- Wang, N., Naruse, K., Stamenović, D., Fredberg, J. J., Mijailovich, S. M., Tolić-Nørrelykke, I. M., Polte, T., Mannix, R., and Ingber, D. E. (2001). Mechanical behavior in living cells consistent with the tensegrity model. *Proceedings of the National Academy of Sciences*, 98(14):7765–7770.
- Watanabe, W., Sata, T., and Ishiguro, A. (2008). An efficient decentralized learning by exploiting biarticular muscles - a case study with a 2d serpentine robot. In *Robotics and Automation, IEEE International Conference*, pages 3826–3831.
- Watson, R. A., Ficici, S. G., and Pollack, J. B. (1999). Embodied evolution: Embodying an evolutionary algorithm in a population of robots. In Angeline, P. J., Michalewicz, Z., Schoenauer, M., Yao, X., and Zalzal, A., editors, *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 335–342, Mayflower Hotel, Washington D.C., USA. IEEE Press.
- Yosinski, J., Clune, J., Hidalgo, D., Nguyen, S., Zagal, J., and Lipson, H. (2011). Evolving robot gaits in hardware: the hyperneat generative encoding vs. parameter optimization. In *Proceedings of the 20th European Conference on Artificial Life, Paris, France*, volume 8, page 12.
- Zykov, V., Bongard, J., and Lipson, H. (2004). Evolving dynamic gaits on a physical robot. In *Proceedings of Genetic and Evolutionary Computation Conference, Late Breaking Paper, GECCO'04*.

# Heterogeneous complexification strategies robustly outperform homogeneous strategies for incremental evolution

Adam Stanton<sup>1</sup> and Alastair Channon<sup>1</sup>

<sup>1</sup>Research Institute for the Environment, Physical Sciences and Applied Mathematics  
Keele University, ST5 5BG, UK  
{a.stanton, a.d.channon}@keele.ac.uk

## Abstract

The evolution of naturalistic, embodied agents and behaviours has been a long-standing goal of Artificial Life since the initial, impressive work of Karl Sims. Incremental evolution has been used extensively to improve the quality of evolutionary search in many complex, non-linear problem spaces. This work sets out to disambiguate the lexicon around incremental evolution, advocating the term *environmental complexification* to represent the complexification of the problem domain. We then go on to analyse various complexification strategies in a structured, complexifiable and yet simple environment: a 3D agent-based obstacle task. We divide the strategies conceptually into *homogeneous* and *heterogeneous*; homogeneous strategies expose successive generations of the population to a single or tightly clustered range of objective functions while heterogeneous strategies present many, covering the range of complexity. It was found that widely-used homogeneous complexification techniques, for example direct presentation of difficult tasks or linearly-increased difficulty, fail due to either loss-of-gradient or temporally-local over-fitting (analogous to *catastrophic forgetting* in neural systems). Heterogeneous methods of complexification (including oscillatory strategies) that eliminate these issues are devised and tested. The heterogeneous category outperforms the homogeneous in all metrics, establishing a much more robust approach to the evolution of naturalistic embodied agents.

## Introduction

The evolution of naturalistic, embodied agents and behaviours has been a long-standing goal of Artificial Life since the initial, impressive work of Karl Sims. We are interested in evolving generalised behaviours rather than those that succeed in only a specific or narrow range of parameters: for example in agents able to climb over arbitrary obstacles rather than just those of a specific (maximal or other) height. It may appear desirable to evaluate each individual in each generation on all combinations of parameters for all behaviours, but this approach is infeasible as the number of combinations scales exponentially. We are therefore interested in evolutionary approaches in which each agent is evaluated on a small subset of parameters, in this paper on a single value for a single behaviour, and yet result in agents able to

perform over the full range of parameters, for example by having evolved generalised behaviours rather than ones that work only in a specific or narrow range of parameters.

Incremental evolution has been used extensively to improve the quality of evolutionary search in many complex, non-linear problem spaces. This work sets out to disambiguate the lexicon around incremental evolution, advocating the term *environmental complexification* to represent the complexification of the problem domain as described above. We then seek to identify and objectively compare the strengths and weaknesses of homogeneous and heterogeneous strategies for complexification of a problem domain when using incremental evolution. In homogeneous complexification strategies, for any short sequence of successive generations the population is exposed to a single or tightly clustered range of objective functions, while heterogeneous strategies present many, covering a range of complexity.

**Incremental Learning in Evolutionary Systems** Inman Harvey's SAGA paradigm, motivated by evolution in the natural world, set the stage for the computational use of incremental evolution by providing an evolutionary mechanism which allows an evolving species to maintain, at least theoretically, most if not all evolutionary pathways as potential candidates for exploration, no matter how converged the population has become to a single point in genotype space (Harvey, 1992, 1997). Once a SAGA algorithm is implemented, objective functions can be changed and the population can be expected to adapt to its new circumstances by traversing *neutral networks* in genotype space (Harvey, 1997, 2001). The requirements for the successful implementation of a SAGA-type incremental process are straightforward: inclusion of mutation as a genetic operator, smooth fitness landscapes and a redundant (high-dimensional) genotype to phenotype mapping which permits neutral networks - interconnected regions of equivalent fitness - to percolate through genotype space. Note that the term *incremental evolution* is used in a sense which implies continued change, development or acquisition of domain knowledge by the algorithm over time. Where there is a gradual increase in difficulty of objective function, we prefer the term *environmental com-*

*plexification* as mentioned in (Mouret and Doncieux, 2009). The label *incremental evolution* is also applied where intermediate solutions are moved to a new objective domain; this case we also consider a flavour of environmental complexification, a point we explain in more detail below.

Some of the earliest work which uses the environmental complexification approach directly is that of Gomez who (in addition to discrete, staged evolution over subtasks) gradually increased the speed of prey in a pursuit-avoidance simulation where neural networks were evolved to control simulated predators (Gomez and Miikkulainen, 1997). This work showed a very large performance gain by using the incremental approach. The work also identified an interesting adaptive approach where complexification is dependent upon agent performance at the current level of complexity. Mouret introduced a more general approach to rewarding sub-task performance than the hand-designed staged approach common until this point (Mouret and Doncieux, 2009). Complex agent behaviour was evolved incrementally in a two-dimensional task in (Robinson et al., 2007) where agents in a discrete world were trained to navigate a hostile environment and ultimately build a strategy for crossing an impassable obstacle. Environmental complexification was used to evolve swarm robots in (Kadota et al., 2012), although the complexification chosen constituted arbitrary, discontinuous changes to the agents' environment and not a smooth transition over a range of difficulties. Notwithstanding, once again the incremental approach delivered a much higher rate of success in the given task (co-operatively foraging for food in a two-dimensional environment). Oh et al. evolved controllers for unmanned aerial vehicles first using a non-incremental strategy. This strategy was found to perform badly as more constraints were added into the objective function so an incremental, task-subdivision strategy was used instead (Oh and Suk, 2013).

### Categorisation of Incremental Learning Techniques

Barlow identified two classes of incremental training schemes: functional incremental evolution and environmental incremental evolution (Barlow et al., 2004). In this definition, functional approaches parameterise fitness functions to increase the apparent difficulty of tasks toward the desired level of complexity whereas environmental approaches modify the environment around the evolving individuals without modifying the fitness function, with the same effect. Sub-categories of incremental evolution identified by Mouret in (Mouret and Doncieux, 2009) are *staged evolution*, *environmental complexification*, *fitness shaping* and *behavioural decomposition*. The most striking of these distinctions, common to both Barlow's and Mouret's work is environmental complexification; this category is of particular interest as semantically it can encompass all of the other categories identified and thus becomes synonymous with the sense of incremental evolution where the problem

is simplified and made progressively more difficult. Additionally, environmental complexification is the only category which adequately encompasses co-evolutionary systems (which can be seen as auto-complexification) which in turn are the natural precursor to open-ended evolutionary systems, the search for which is an active area of research in the Artificial Life domain.

**Incremental Learning in Neural Systems** The idea of incremental learning is not confined to evolutionary adaptive algorithms; neural network research has also considered this both as a problem (learning invariances piecewise) and as a solution (tackling complex problems) for networks generally, outside of any particular training scheme. In the standard approach of using neural networks, training and application are distinct phases: all training data are presented to the network and the system learns the invariances and abstractions in that data using some learning algorithm. Then, this trained network is put to work on unseen data. This method of presentation can make it difficult for the network to adapt to new, unseen data at a later time and cause networks to suffer the phenomenon of *catastrophic forgetting* (McCloskey and Cohen, 1989). In contrast to this, incremental learning algorithms are designed to allow the neural system to continually adapt to new information whilst maximising the information available in the network from previous training. This is an important concept for real world applications as often data is not available all at once and sometimes learning guides further exploration, meaning that learning is a continuous process rather than a discrete activity (Giraud-Carrier, 2000). One popular solution to the catastrophic interference problem found in these incremental learning schemes is to *rehearse* either already known data or pseudo-data representing the knowledge already in the network, interleaved with training on new information. See (French, 1999) for an overview and e.g. (Guajardo et al., 2010) for recent work using this technique.

**Complexification Strategies** As noted above, previous work has successfully leveraged the power of incremental evolution through successive increases in environmental complexity. However, attention has been focused solely on the outcome and the particular strategies used to complexify the environment have not been examined in detail. We assert that a rigorous theoretical underpinning of complexification is necessary both for the practical application of incremental evolution and the further elucidation of the interplay of agent and environment in co-evolutionary settings which ultimately lead to unbounded evolutionary activity.

The naive strategy presents the most difficult task to the evolving species at every opportunity. This *straw man* is unlikely to be successful: it was the failing of this approach that spurred the development of alternative, progressive strategies, e.g. linear increase in difficulty as time passes.



The linear approach has been used often to circumvent the bootstrapping problem, one of the first attempts occurring in (Gomez and Miikkulainen, 1997). This approach is a natural extension of human learning - start easy and then get harder - and the simplicity of implementation and broadness of potential application strengthen its appeal. Many task decomposition strategies can also really be considered an implementation of a linear increase in complexity, albeit discrete rather than continuous. Gomez also proposed an extension to the linear increase in task complexity where difficulty is only increased when the evolving species achieves a certain level of performance against the current objective function. This interesting strategy has not been developed in detail by others but we consider it a good candidate for analysis as it enforces gradient at every level of difficulty, potentially solving some or all of the issues described in the introduction to this work.

Although not often described in previous work, random presentation of different task complexities may also be useful and finally, drawing upon the ideas of incremental learning in neural systems we propose a strategy of repeated presentation of earlier, simpler tasks in an evolutionary setting. These strategies may have something to offer beyond linear or adaptive monotonic changes in task complexity.

## Hypotheses

We anticipate that homogeneous complexification strategies, for example direct presentation of difficult tasks or linearly-increased complexity, will perform poorly due to either loss-of-gradient or temporally-local over-fitting (analogous to *catastrophic forgetting* in neural systems). Heterogeneous strategies are our proposed approach to overcoming forgetting, as an analogue of rehearsal, with smoothly changing heterogeneous strategies, such as oscillatory strategies, also overcoming the loss-of-gradient problem. For oscillatory strategies, the current range of difficulties is from zero to the amplitude of oscillation. A gradual increase of this range may be expected to show improved performance. At very low frequencies, such a strategy would degenerate to the homogeneous linear strategy, and at very high frequencies to the random strategy. Thus, we propose the following hypotheses:

H1: Homogeneous strategies will fail to achieve good coverage on the evaluation task.

H2: Heterogeneous strategies (with the possible exception of random) will achieve better coverage than homogeneous strategies.

H3: Heterogeneous strategies with a range of difficulties increasing over time will outperform heterogeneous strategies with constant range.

H4: A heterogeneous strategy using an oscillatory approach, as an analogue of rehearsal, will exhibit an optimal frequency for any particular problem.

## Method

The general setup of our experiment is designed to test the above hypotheses in a task which provides a smooth fitness landscape and neutrality in genotype space. We have chosen the evolution of controllers for three-dimensional agents as the platform, tasked with learning how to walk and climb over an obstacle. The height of the obstacle represents the ‘complexification’ parameter of the system; task difficulty varies somewhat as obstacle height varies but the ultimate objective for the agents is to deal with every possible obstacle - this is the most complex case. Thus, we can assess which of many possible complexification strategies (that is presentation of tasks of various difficulties) provide the strongest gradient for the evolutionary system to climb and the most robust final evolved agents.

**A. The Physical Model** In the tradition founded by (Sims, 1994) and continued by many others, we perform all experiments on agents in a three-dimensional virtual world consisting of collidable rigid bodies connected by powered constraints. Unlike Sims, our morphology is a fixed quadruped which is controlled by a feed-forward three-layer perceptron augmented by sinusoidal input. The cuboid quadruped torso (length 0.4m) is supported by four limbs, each comprising an upper and lower portion (length 0.2m). Constraints with two degrees of freedom limit the motion of torso and upper limb at the hips; constraints with one degree of freedom limit the motions of lower limb and upper limb at the knee. (See figure 1 for a visual representation.) The range of motion in each case is limited to  $\pi$  radians centred on the diagonals extending from the centre of the agent to the lower corners, also the points of attachment for each limb. The maximum power that can be applied at any constraint is a force of  $\pm 0.1N$ . The obstacle is situated 1m from the agent’s origin and extends to infinity in  $x$  and for 0.02m in  $y$ . The height of the obstacle is varied as described elsewhere. The physical simulator used was ODE 0.12, using double-precision arithmetic, the standard big-matrix step function and a step-size of 0.02s. Coulomb friction was applied at contacts between the agent, the obstacle and the ground plane with  $\mu = +\infty$ .

**B. The Control System** The agent controller is modelled by a standard three-layer feed-forward neural network with 12 hidden nodes. Networks receive 4 real-valued inputs in addition to 12 joint-angle sensors. Inputs comprise two sinusoidal oscillators (sine and cosine, period 1 second), an input describing the target location in relation to agent position and orientation (difference between distance from target to each ear, divided by distance between ears) and an up-sensor which describes the orientation of the agent’s head relative to the ground plane. Network updates are made synchronously with physics integration. Each hidden node activation is a weighted sum of its inputs with a hyperbolic tan activation. Each output node activation is a weighted sum of hidden nodes with a logistic activation function.

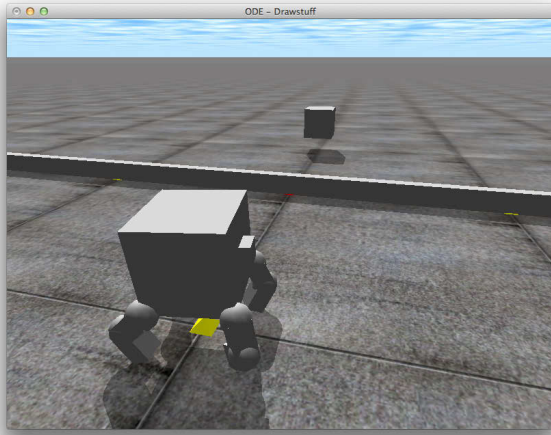


Figure 1: Visualisation of physical environment. Agent, obstacle and target location are shown.

**C. The Evolutionary Algorithm** Individual genotypes specify floating-point weights for the neural control system. Initial values for the first generation are drawn from a uniform distribution  $x \in [-1, 1]$ . In each run, the evolutionary simulation is progressed for 5000 generations using a population of 50 individuals. Individuals are evaluated for 20 simulated seconds and the objective function is defined as the total distance covered in the x-y plane toward a target position situated on the other side of the obstacle. At each new generation, individuals are scored according to the objective function and ranked in order of fitness. The lower half of the population is replaced with mutated, crossed-over variants of the upper half. Mutation occurs on average twice per genotype and consists of adding a value drawn from a Gaussian distribution with  $\sigma = 1$  and  $\mu = 0$ . Single-point crossover is implemented at a random point on the genotype and crosses the current parent individual with another random individual from the best half of the population (possibly itself.)

**D. The Experimental Setup** Sixteen possible strategies for environmental complexification have been identified and tested; each of these strategies modifies the height of the obstacle in the environment for the current generation of the species. In every case the maximum height of the obstacle,  $\tau$  is 0.1m. Height function  $h$  for generation  $G$  and wavelength  $\lambda$  is defined for each strategy as follows:

1. Direct presentation of environment with complexity  $\tau$  at every generation:  $h(G) = \tau$  (Strategy 1)
2. Presentation of a randomly complex environment at each generation, with complexity drawn from a uniform distribution between 0 and  $\tau$ :  $h(G) = \text{random}(0, \tau)$  (Strategy 2)

3. Gradual complexification of the environment, with complexity interpolated linearly between 0 and  $\tau$  from generation 0 to generation 4000 and fixed at  $\tau$  from generation 4001 to 5000 (Strategy 7):

$$h(G) = \begin{cases} \frac{\tau G}{4000}, & G < 4000, \\ \tau & \text{otherwise} \end{cases}$$

4. Oscillating complexification of the environment ( $\lambda = 50, 100, 200, 400$  generations), with complexity following a sinusoidal increase and decrease over wavelength  $\lambda$  with maximum amplitude  $\tau$  (Strategies 3, 4, 5 and 6):

$$h(G, \lambda) = \tau \frac{1 + \sin(\frac{2\pi G}{\lambda} - \frac{\pi}{2})}{2}$$

5. Oscillating complexification of the environment as above, with maximum amplitude interpolated linearly between 0 and  $\tau$  from generation 0 to generation 4000 and fixed at  $\tau$  from generation 4001 to 5000 (Strategies 8, 9, 10 and 11):

$$h(G, \lambda) = \begin{cases} \frac{\tau G}{4000} \frac{1 + \sin(\frac{2\pi G}{\lambda} - \frac{\pi}{2})}{2}, & G < 4000, \\ \tau \frac{1 + \sin(\frac{2\pi G}{\lambda} - \frac{\pi}{2})}{2} & \text{otherwise} \end{cases}$$

6. Adaptive modification of 1, where  $\tau$  is increased by 1% when the average fitness of the population has increased or remained the same and decreased by 1% if average fitness has decreased. (Strategy 12)
7. Adaptive modification of 5 where  $\tau$  is increased by 1% when the average fitness of the population has increased or remained the same and decreased by 1% if average fitness has decreased. (Strategies 13, 14, 15 and 16).

## Results

Table 1 shows that *no homogeneous complexification strategy (direct, linear or adaptive) was able to achieve success on all task difficulties*, in any experimental run. In contrast, *all heterogeneous strategies did*. The adaptive oscillating ( $\lambda=50$ ) strategy achieved 100% success in 20% of runs and 95% success in 48% of runs.

Figure 2 shows a complete view for each strategy, with  $\lambda=50$  selected for each oscillating strategy and each strategy's 100 runs sorted along the horizontal axis by proportion of successful evaluations (shown on the vertical axis). Note that we are primarily interested in the upper portion of this graph, that is in those populations able to complete the task at most obstacle heights). The adaptive strategy generated fewer populations than the linear strategy, successful on fewer than 50% of evaluations (over the full range of obstacle heights) but of greater interest is that it generated only a comparable number of populations successful on more than 90% of evaluations. The random strategy, whilst better than all homogeneous strategies, is

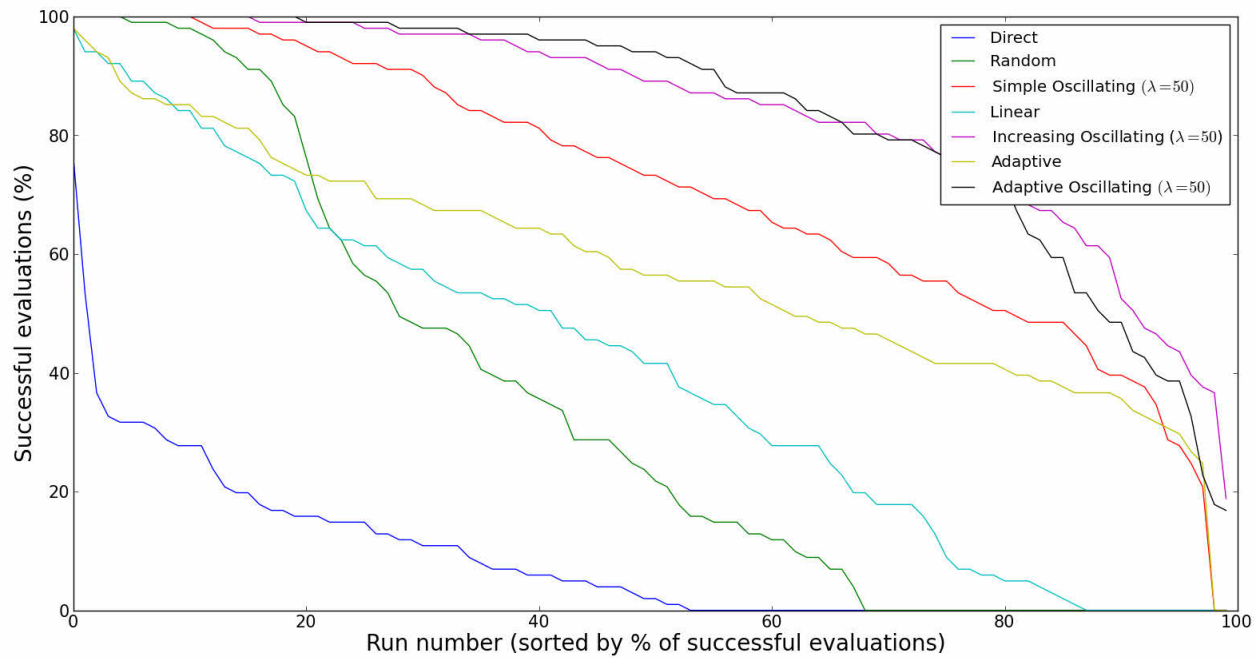


Figure 2: Performance of various strategies, 100 runs per strategy sorted best to worst.

| Number                   | Strategy                                 | % runs with success of at least: |      |
|--------------------------|------------------------------------------|----------------------------------|------|
|                          |                                          | 95%                              | 100% |
| Homogeneous Strategies   |                                          |                                  |      |
| 1                        | Direct                                   | 0%                               | 0%   |
| 7                        | Linear                                   | 1%                               | 0%   |
| 12                       | Adaptive                                 | 2%                               | 0%   |
| Heterogeneous Strategies |                                          |                                  |      |
| 2                        | Random                                   | 13%                              | 5%   |
| 3                        | Simple Oscillating ( $\lambda=50$ )      | 21%                              | 11%  |
| 4                        | Simple Oscillating ( $\lambda=100$ )     | 16%                              | 7%   |
| 5                        | Simple Oscillating ( $\lambda=200$ )     | 17%                              | 8%   |
| 6                        | Simple Oscillating ( $\lambda=400$ )     | 10%                              | 2%   |
| 8                        | Increasing Oscillating ( $\lambda=50$ )  | 39%                              | 16%  |
| 9                        | Increasing Oscillating ( $\lambda=100$ ) | 30%                              | 14%  |
| 10                       | Increasing Oscillating ( $\lambda=200$ ) | 30%                              | 12%  |
| 11                       | Increasing Oscillating ( $\lambda=400$ ) | 29%                              | 10%  |
| 13                       | Adaptive Oscillating ( $\lambda=50$ )    | 48%                              | 20%  |
| 14                       | Adaptive Oscillating ( $\lambda=100$ )   | 44%                              | 16%  |
| 15                       | Adaptive Oscillating ( $\lambda=200$ )   | 26%                              | 9%   |
| 16                       | Adaptive Oscillating ( $\lambda=400$ )   | 31%                              | 9%   |

Table 1: Number of runs achieving success on 95% and 100% of obstacle heights.

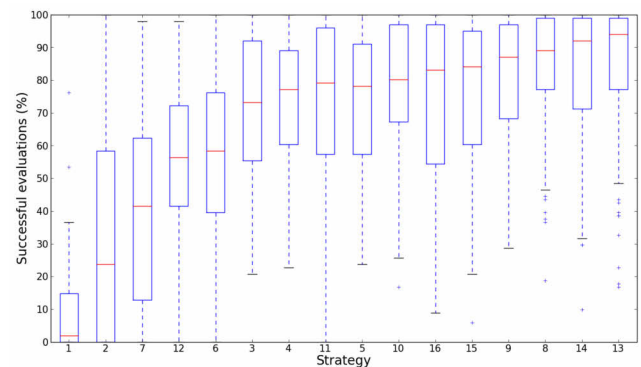


Figure 3: Aggregate success rate over all obstacle heights for various strategies, sorted by median success rate. Each evolved population was evaluated on the task at heights 0%, 1%, ... 100%. (See Table 1 for description of the numerical labels.)

|    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|
| 1  |   | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑  | ↑  | ↑  | ↑  | ↑  | ↑  | ↑  |
| 2  | ← |   | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑  | ↑  | ↑  | ↑  | ↑  | ↑  | ↑  |
| 3  | ← | ← |   |   |   | ← | ← | ↑ | ↑ | ↑  |    | ←  | ↑  |    |    |    |
| 4  | ← | ← |   |   |   |   |   | ↑ | ↑ | ↑  |    |    | ↑  | ↑  |    |    |
| 5  | ← | ← |   |   |   | ← | ← | ↑ | ↑ | ↑  |    | ←  | ↑  | ↑  |    |    |
| 6  | ← | ← | ↑ | ↑ | ↑ |   | ← | ↑ | ↑ | ↑  | ↑  |    | ↑  | ↑  | ↑  | ↑  |
| 7  | ← | ← | ↑ | ↑ | ↑ | ↑ |   | ↑ | ↑ | ↑  | ↑  | ↑  | ↑  | ↑  | ↑  | ↑  |
| 8  | ← | ← | ← | ← | ← | ← | ← |   |   | ←  | ←  | ←  |    |    | ←  | ←  |
| 9  | ← | ← | ← | ← | ← | ← | ← |   |   |    | ←  | ←  | ↑  |    |    |    |
| 10 | ← | ← | ← | ← | ← | ← | ← | ↑ |   |    |    | ←  | ↑  | ↑  |    |    |
| 11 | ← | ← |   |   |   | ← | ← | ↑ | ↑ |    |    | ←  | ↑  | ↑  |    |    |
| 12 | ← | ← | ↑ | ↑ | ↑ |   | ← | ↑ | ↑ | ↑  | ↑  |    | ↑  | ↑  | ↑  | ↑  |
| 13 | ← | ← | ← | ← | ← | ← | ← |   | ← | ←  | ←  | ←  |    |    | ←  | ←  |
| 14 | ← | ← | ← | ← | ← | ← | ← |   |   | ←  | ←  | ←  |    |    | ←  | ←  |
| 15 | ← | ← |   |   |   |   |   |   |   |    |    |    | ←  | ↑  | ↑  |    |
| 16 | ← | ← |   |   |   | ← | ↑ |   |   |    |    |    | ←  | ↑  | ↑  |    |

Table 2: Significance table showing one-tailed statistical relationship between strategies,  $p < 0.05$ . Arrows indicate statistical dominance of one strategy over another (significantly higher median value). Statistical test used was the Mann-Whitney U-test.

by far the worst method of the heterogeneous strategies. In turn, the simple oscillating strategy is outperformed by the increasing oscillating and adaptive oscillating strategies.

Figure 3 shows a box plot of successful evaluations (%) for each strategy (with whiskers to 1.5 interquartile ranges below and above the lower and upper quartiles), complete with a range of wavelengths (50, 100, 200 and 400 generations). Mann-Whitney U tests were performed to examine significant differences in median number of evaluative successes between strategies and within strategies (by varying wavelength). In table 2, a left arrow indicates that the strategy corresponding to the row number has a significantly higher ( $p < 0.05$ ) median success rate than the strategy corresponding to the column number (and an up arrow vice versa), shown particularly clearly by strategies 8 and 13. Within each of the increasing and adaptive oscillating strategies, the median number of successful evaluations was found to be significantly higher ( $p < 0.05$ ) for strategies with wavelengths of 50 to 100 generations when compared to the same strategy with four times the wavelength or higher. Within the simple oscillating strategy, the long wavelength (400 generations) produced a significantly lower median ( $p < 0.05$ ) than shorter wavelengths (50, 100, 200 generations).

Strategies which oscillate showed the best performance. We found no significant difference in median between the increasing and adaptive oscillating strategies at equal wavelengths. We found that either a linear or an adaptive increase in maximum amplitude over the training time performed significantly better than simple oscillation. For both increasing oscillating and adaptive oscillating the two lower

wavelengths (50 and 100 generations) showed a significantly higher ( $p < 0.05$ ) median number of successful evaluations than the simple oscillating strategies at all wavelengths.

On average, the adaptive strategy performed significantly better than the direct, linear and random strategies, and significantly worse than every oscillatory strategy (except the simple oscillating strategy at wavelength 400 for which we found no significant difference).

The linear strategy resulted in a significantly higher median number of successful evaluations than the direct and random strategies (even though the random strategy produced more highly fit populations from many more runs) and a significantly lower median than all other strategies.

On average, the random strategy performed significantly worse than all other strategies except for the direct method, which was significantly worse than all other strategies.

In order to determine whether the poor results of the linear strategy is due to either evolutionary loss or failure to gain we determined the proportion of successful evaluations at each obstacle height throughout the evolutionary progress, for each strategy. Figure 4 shows that all strategies achieved 8% success at all obstacle heights, with the exceptions of direct (for which obstacle height is always 100%) and adaptive (low coverage at high obstacle height). The linear strategy achieved more successful evaluations than the simple oscillating strategy at all wavelengths during the evolutionary phase, indicating that its ultimate failure is due to evolutionary loss rather than a failure to gain. Only 10% of the final population from linear runs were able to walk to the target with no obstacle, compared to at least 69% for the increasing and adaptive oscillating strategies. As in figure 3, figure 5 shows the number of successful evaluations for each strategy but drawn only from those runs able to reach the target with no obstacle (that is eliminating those runs which experienced the greatest evolutionary loss), and shows that in these cases, linear performance has a range comparable to the simple oscillatory strategies and a median comparable to the increasing and adaptive oscillating strategies.

To investigate the dependency of success rate on oscillatory frequency we evaluated the simple, increasing and adaptive oscillating strategies across as range of wavelengths from 2 to 10000 generations; figure 6 demonstrates this relationship. As wavelength approaches zero, the proportion of successful evaluations approaches that of random. As wavelength approaches total evolutionary time (number of generations), the proportion of successful evaluations approaches that of linear. Between these points, it can be seen that for each strategy there is an optimal wavelength (for the current algorithm, around 50-100 generations).

## Discussion

It is clear from the results presented above that there is a strong distinction between the homogeneous and heterogeneous strategies. No homogeneous strategy achieved 100%



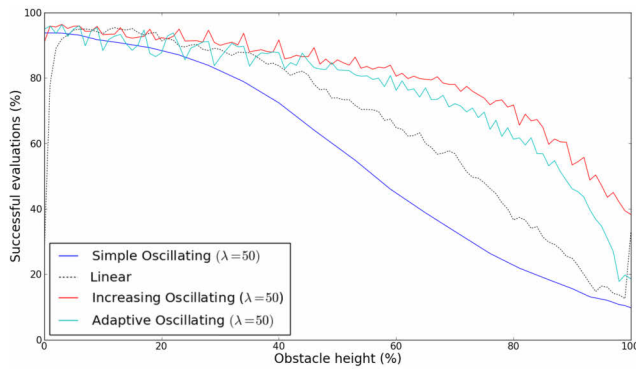


Figure 4: Strategy performance against obstacle height during evolution.

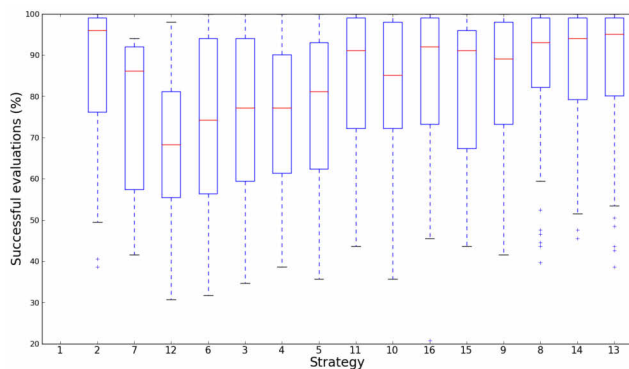


Figure 5: Success rate over all obstacle heights for various strategies (only aggregates runs which solved the task at zero-height). (See Table 1 for description of numerical labels; order preserved from Figure 3.)

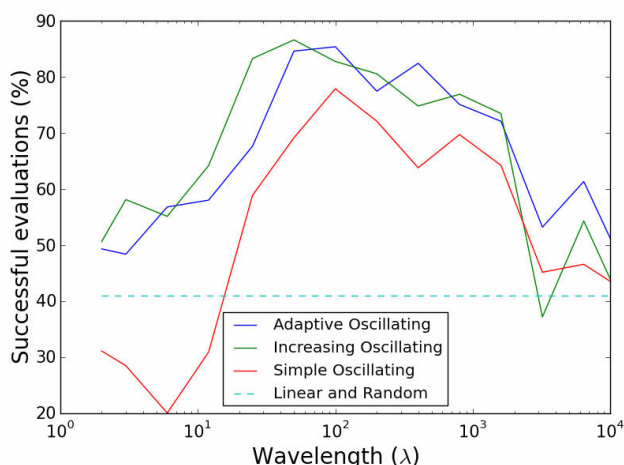


Figure 6: Strategy performance (% success) against wavelength for oscillating strategies.

coverage of the evaluation task in any run (Table 1) whereas all heterogeneous strategies did. Within the homogeneous category, the trivial, direct method of presentation was by far the least successful (Figures 2 and 3). The linear strategy was more successful but the best strategy in this category was the adaptive strategy. The poor performance of the homogeneous category can be explained by *evolutionary forgetting*: these strategies have either lost evolutionary gradient and drifted away from any early successes (linear) or over-specialised on later parts of the problem (adaptive).

The heterogeneous strategies perform better than the homogeneous group: the most successful strategies we explored all made multiple presentations of easier tasks at later stages of the evolutionary run, at the expense of fewer presentations of later tasks. These strategies all performed well at the hardest task and had the best generalisation performance over the whole range of tasks, suggesting that our hypothesis has merit.

The random strategy is the least successful strategy in this category. This may be due to the same problem of gradient loss as in the homogeneous group. As found in the homogeneous group, the linear and adaptive modifications of the oscillating strategy showed the best performance of all; the slow increase in task difficulty maintains a strong evolutionary gradient and the cyclical nature of task presentation consolidates earlier gains and causes the evolving population to prefer generalised solutions abstracted over the whole problem domain.

This consolidation is dependent on the frequency of representation of earlier, or easier, parts of the task. When investigating this frequency, it can be seen that a clear optimum exists in the frequency domain where cyclical strategies are able to maximise this consolidation without losing gradient. This optimum is likely to be problem-specific and a range of values should be explored for any given task. However, in the limit of wavelength, i.e. at very low and very high frequencies, it can be seen that the performance of the evolving populations begins to approximate, for low and high frequencies respectively, the linear and random strategies. This offers an abstract insight into the underlying mechanism at work - the maintenance of selective pressure and whole-task capability. As these components reduce in effectiveness due to the change in wavelength, so the evolving populations degenerate into the simpler strategies described above. The successful cases are those where environmental change is fast enough to induce a generalisation in the agent's approach to the task but slow enough to prevent catastrophic loss of gradient when evaluating partial solutions.

## Conclusions

The points made in the discussion section support our hypotheses. The homogeneous strategies showed weak performance on the evaluation task, with no strategy achiev-

ing full coverage in any run. Conversely the heterogeneous strategies, including surprisingly the random strategy, all achieved full coverage in some runs. Those heterogeneous strategies with a range of difficulties increasing over time (increasing and adaptive oscillating) outperformed the simple (constant range) oscillating strategies, showing a much higher proportion of successful runs. Finally, we demonstrated that oscillating strategies do exhibit an optimal frequency.

Complexification strategies for incremental evolution offer a powerful mechanism for adaptive problem solving. However, this power comes at a price: it is easy to lose information learned earlier in the process. In order to fully exploit this power appropriate complexification strategies have to be realised in order to drive populations along desirable adaptive pathways. There are many options for formulating these strategies: much previous work has involved, in one manner or another, a simplification of the objective function and then a progressive complexification as time passes. In this work we found that many strategies encounter loss-of-gradient or over-fitting problems. We present a solution in the form of heterogeneous complexification strategies which combine solutions to those problems to deliver robust populations. Our approach can be translated to many scenarios where progressive complexification is used to guide an incremental evolutionary process; further exploration of the limitations and advantages of heterogeneous complexification within different problem domains would be useful in order to generalise these conclusions. Additionally, the oscillating strategies exhibited an optimal wavelength for representation. It is unclear whether this optimum is task-dependent or whether there is an underlying principle and optimal wavelength for this type of training; this question also merits further work.

Finally, we would advise that in general while a random presentation of subtasks or objective difficulty levels is preferable to a linear increase, as a minimum guideline an increasing heterogeneous complexification strategy should be used. This rehearsive, cyclical approach to presentation not only maintains evolutionary gradients but also promotes generalisation amongst the evolving populations from subtask-specific adaptation to performance across the super-task.

## References

- Barlow, G., Oh, C., and Grant, E. (2004). Incremental evolution of autonomous controllers for unmanned aerial vehicles using multi-objective genetic programming. In *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, volume 2, pages 689–694.
- French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.
- Giraud-Carrier, C. (2000). A note on the utility of incremental learning. *AI Communications*, 13(4):215–223.
- Gomez, F. and Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:5–317.
- Guajardo, J. A., Weber, R., and Miranda, J. (2010). A model updating strategy for predicting time series with seasonal patterns. *Applied Soft Computing*, 10(1):276–283.
- Harvey, I. (1992). Species adaption genetic algorithms: A basis for a continuing saga.
- Harvey, I. (1997). Artificial evolution for real problems. In Gomi, T., editor, *Proceedings of the 5th International Symposium on Evolutionary Robotics*.
- Harvey, I. (2001). Artificial evolution: A continuing saga. In *Proceedings of the International Symposium on Evolutionary Robotics From Intelligent Robotics to Artificial Life*, ER '01, pages 94–109, London, UK, UK. Springer-Verlag.
- Kadota, T., Yasuda, T., Matsumura, Y., and Ohkura, K. (2012). An incremental approach to an evolutionary robotic swarm. In *System Integration (SII), 2012 IEEE/SICE International Symposium on*, pages 458–463.
- McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *The psychology of learning and motivation*, 24(109-164).
- Mouret, J.-B. and Doncieux, S. (2009). Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *IEEE Congress on Evolutionary Computation*.
- Oh, S.-H. and Suk, J. (2013). Evolutionary controller design for area search using multiple uavs with minimum altitude maneuver. *Journal of Mechanical Science and Technology*, 27(2):541–548.
- Robinson, E., Ellis, T., and Channon, A. (2007). Neuroevolution of agents capable of reactive and deliberative behaviours in novel and dynamic environments. In *Advances in Artificial Life*, pages 345–354. Springer.
- Sims, K. (1994). Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques, SIGGRAPH '94*, pages 15–22, New York, NY, USA. ACM.

# Conformity and Nonconformity in Collective Robotics: A Case Study

Gregory Vorobyev\*, Andrew Vardy, and Wolfgang Banzhaf

Memorial University of Newfoundland, St. John's, Canada, \*gvorobyev@mun.ca

## Abstract

In this work, we develop a social behavioral model designed for multi-agent systems for solving the collective sorting task. Experiments show that under this model agents are capable of improving their performance significantly and can achieve better results than conventional swarms of agents lacking communication and social abilities.

## Introduction

In his fascinating book “The Social Animal”, Elliot Aronson defines conformity “as a change in a person’s behavior or opinions as a result of real or imagined pressure from a person or group of people” (Aronson, 2007). Conformity is one of the essential and most important aspects of human society. Failure to conform to the rules issued by society may turn out to be not only merely inconvenient, but even dangerous. Driving down the wrong side of the street can be an example of a nonconformist behavior that will most likely lead to tragic consequences (Aronson, 2007). Nonconformity, however, often works to the long-term benefit of the society as a whole. One example of “useful” nonconformity could be a scientist attempting to look at a well-known problem from an entirely new angle, which can be controversial to the viewpoint generally accepted. Sometimes this could result in a revolution in science (i.e., consider Einstein’s theory of relativity vs. classical Newtonian mechanics).

In this paper, we develop a simple model of conformity and nonconformity in an artificial society. This model is not intended to be a valid counterpart of the relevant phenomena in the human society. Rather, our aim is to attempt to incorporate some degree of *social intelligence* (specifically, the ability to choose between conformity and nonconformity in behavior) in artificial agents and to study whether this additional “social” part of agent reasoning could be beneficial in terms of the performance on the task being executed.

As a case study, we use the sorting task in the context of multi-agent systems, which is formulated as follows: given a set of objects of different types  $\{x_1, x_2, \dots, x_n\}$ , the group of  $N$  agents is to collect them into homogeneous clusters. *Swarm robotics* offers distributed algorithms for solving this

problem (Bonabeau et al., 1999; Deneubourg et al., 1991; Bayindir and Sahin, 2007; Beckers and Holland, 1994; Melhuish and Hoddell, 1998; Wang and Zhang, 2004; Verret et al., 2004; Vorobyev et al., 2012). The distinguishing property of the swarm-based approach is that agents operate and perceive only locally; thus, no global supervision and/or knowledge is required. While swarm-based algorithms typically show slower convergence than centrally-controlled approaches, their advantages are simplicity, flexibility and robustness (Sahin, 2005). A swarm agent has very limited sensing capabilities. As it is highlighted in the next section, the only input to one popular swarm-based sorting algorithm by (Deneubourg et al., 1991) is  $f(x)$ , which roughly estimates the density of objects of type  $x$  in the immediate neighborhood. To obtain  $f$ , an agent only needs a sensor which allows to recognize the type of an object, if there is any, right in front of the agent. In many contributions in this field, agents are even not aware of each other; i.e., *kin recognition* is not present (Bayindir and Sahin, 2007). Such “social ignorance” is viewed as beneficial, because it guarantees scalability and robustness of the approach.

This work primarily concentrates on extending Deneubourg et. al.’s sorting algorithm by introducing additional input information. We refer to this information as *social*, because it represents knowledge about the goals of other agents. Obtaining this new type of information will require explicit communication, as opposed to implicit communication, commonly employed by robotic swarms (e.g., through *stigmergy* (Beckers and Holland, 1994)). Since it is generally accepted that communication capabilities in swarm-based systems should be kept as minimal as possible, we also refer to the group of our “socially intelligent” agents as a *society* as opposed to a *swarm*. This work offers some evidence that socially aware agents could perform more effectively and “intelligently” than their swarm counterparts. The term *social intelligence*, as well as *artificial social intelligence* are probably too broad to cover in one paper; rather, we concentrate on just one social phenomenon – conformity and nonconformity. Much like traditional swarm robotics is inspired by social insects (see,

e.g., (Bonabeau et al., 1999)), we are inspired by arguably the most successful social beings we know – humans.

Although sharing information between agents as such is not new in the field of collective and swarm robotics, most of the contributions that employ this concept tend to focus on extending individual's sensory capabilities by accessing perception, or memory, of others (Verret et al., 2004; Grech et al., 2012). For example, in the collective clustering task, agents can share information about the clusters they have seen, and perform different actions based on that information; in some sense, this would be equivalent to having non-communicating agents with enhanced sensing capabilities (e.g., increased sight range). In contrast, this paper studies how agents with limited sensing capabilities make decisions based purely on the number of other agents that made a similar (or different) decision. *Conformist* agents always tend to make decisions that are similar to the course of the majority, whereas *nonconformist* agents are more independent. In some sense, our agents resemble *zero-intelligent* particles described in (Bentley and Ormerod, 2011).

One important assumption made in this paper, in addition to the ability of agents to distinguish objects of different types and to explicitly communicate with each other, is that they can remember *home locations* in the environment to which they can return if they wish so. This assumption is, in fact, biologically plausible. For example, honeybees can travel long distances and return to their hives (Seeley, 2010). We are not concerned with the details of the implementation of a homing mechanism; rather, we just assume that our agents can store their home locations in their *local coordinate system*. This information will further be subject to social exchanges.

The next section describes the sorting algorithm followed by socially intelligent agents. We then present experimental results, a comparison with Deneubourg et. al.'s "socially ignorant" agents, and the analysis of different parameters of the social model.

## The model

The proposed "social" algorithm is derived from Deneubourg et. al.'s model, first introduced in 1991 (Deneubourg et al., 1991) (which from now on is referred to as "Ant-Like Robots", or "ALR", model). The behavior of ALR agents can be summarized as follows. Each agent moves randomly. If an agent who is not carrying an object encounters an object of type  $x$ , it decides whether or not to pick it up. The probability  $p_p(x)$  of doing so is defined as follows:

$$p_p(x) = \left( \frac{k_p}{k_p + f(x)} \right)^2 \quad (1)$$

where  $p_p(x)$  is the probability to pick up the object of type  $x$ ,  $0 \leq f(x) \leq 1$  is a function estimating the relative density

of objects of type  $x$  in the current neighborhood, and  $k_p$  is an arbitrary constant. Each agent has a short-term memory  $m$  of size  $N_m$  for storing the information of what kind of objects (if any) it has encountered in the recent past.  $f(x)$  is calculated based on that memory:

$$f(x) = \frac{1}{N_m} \sum_{i=1}^{N_m} \begin{cases} 1, & \text{if } m_i \equiv x, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

In a similar manner, the probability of depositing the object being carried upon encountering an empty cell is defined as follows:

$$p_d(x) = \left( \frac{f(x)}{k_d + f(x)} \right)^2 \quad (3)$$

where  $k_d$  is a constant. Thus,  $p_p(x)$  decreases with  $f(x)$  from 1 (when  $f(x) = 0$ ) to 0.25 (when  $f(x) = k_p$ ), and  $p_d(x)$  increases with  $f(x)$  from 0 (when  $f(x) = 0$ ) to 0.25 (when  $f(x) = k_d$ ).

## Home locations and division of labor

In the model described, agents pick up and put down objects as they walk randomly in the environment. The performance of the sorting task, however, can be significantly improved if each agent has a *home location*, that is, the location where the cluster of objects is to be formed. The algorithm can then be modified as follows. An agent starts looking for an object to pick up by roaming randomly. Upon encountering an object, the agent picks it up with probability  $p_p$ . The agent then deterministically returns to its home location. Once the home location is reached, the agent starts roaming randomly and tries to put down the object into any empty cell it finds with the probability  $p_d$ . When the object is deposited successfully, the agent starts looking for another object.

It is obvious that bringing objects of different types to the same home location will not solve the sorting problem. Thus, there must be only one type of object associated with each particular home location and hence with each agent. To be more general, from now on, we will refer to the object type as *task*; e.g., "agent  $A$  is executing task  $x$ " is equivalent to "agent  $A$  looks for objects of type  $x$  and brings them to its home location". The question is then how to assign tasks to agents (that is, how to configure *division of labor*). One simple solution (perhaps not optimal, but acceptable in our case) would be to assign a task to an agent according to the type of the very first object which that agent encounters at the beginning of the simulation. Another question is how to assign initial home locations to agents; in this paper, this assignment is uniformly random.

As it will be shown further, a group of agents employing the homing algorithm demonstrates better performance on average than randomly roaming ALR agents. It is obvious, however, that the homing approach itself has a significant drawback: it is not flexible. The assignment of tasks and



home locations to agents is fixed. Therefore, if none of the agents has been assigned to a task  $x$ , then objects of that type will be unaffected by the sorting process. On the other hand, if more than one agent have been assigned to task  $x$ , then convergence to a single cluster of type  $x$  will never be achieved (assuming that the home locations of  $x$ -agents are sufficiently far from each other). The situation is even worse if the number of agents is large, or the distribution of objects of different types is not uniform.

### Conformity and nonconformity

We propose to solve the problem of a fixed assignment of home locations and tasks by using explicit communication between agents. If two agents  $A$  and  $B$  are currently located within the communication range  $r$  of each other, they can share information about their home locations (denoted as  $h_A$  and  $h_B$ , respectively) and tasks (denoted as  $x_A$  and  $x_B$ ).

If both of them are working on the same task  $x$ , then they should agree on a single home location to guarantee convergence to a single cluster. In our model, the probability  $p(h_B \leftarrow h_A)$  of that  $h_A$  will convert to  $h_B$  is defined as follows:

$$p(h_B \leftarrow h_A) = \min \left[ 1, \left( \frac{|h_B|}{|h_A| + l_h} \right)^2 \right] \quad (4)$$

where  $0 \leq |h_X| \leq 1$  is the estimated proportion of other agents (excluding  $X$ ) that have their home locations at  $h_X$ , and  $l_h$  is a constant, which we interpret as *home loyalty*. Thus,  $p(h_B \leftarrow h_A)$  increases with  $|h_B|$  and decreases with  $|h_A|$ .

$|h_X|$  can be estimated as follows. Each agent  $A$  has a memory  $M$  containing information about other agents met (agents have unique identifiers associated with them). If a piece of that memory  $M_B$  contains information about some other agent  $B$ , then  $M_B^h$  will denote  $h_B$  at the time of the last conversation between  $A$  and  $B$ , and  $M_B^x$  will denote  $x_B$  at the same moment in time. Note that  $M_B^h$  at any given moment is not necessarily equal to the current  $h_B$ , because  $B$  might have changed its home position since the last time  $A$  and  $B$  communicated. Then  $A$  can estimate  $|h_X|$  using the following formula:

$$|h_X| = \frac{1}{|M|} \sum_{i=1}^{|M|} \begin{cases} 1, & \text{if } M_i^h \equiv h_X, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Note that  $M_i^h$  and  $h_X$  should be defined with respect to the same reference frame, for example, with the local coordinate system associated with  $A$ . Thus, whenever  $B$  informs  $A$  about its home location  $h_B$  defined with respect to  $B$ 's reference frame,  $A$  should transform this vector to  $A$ 's coordinate system. We assume that  $A$  is able to perform this operation by using the information about the location of  $B$  with respect to  $A$  at the time of communication.

If during a conversation  $A$  and  $B$  discover that they work on different tasks, there is a chance that one of them will convert to the other's task. The probability of doing so  $p(x_B \leftarrow x_A)$  is defined similar to that of converting to the other's home location, namely:

$$p(x_B \leftarrow x_A) = \min \left[ 1, \left( \frac{|x_B|}{|x_A| + l_x} \right)^2 \right] \quad (6)$$

where  $0 \leq |x_X| \leq 1$  is the estimation of the number of other agents (excluding  $X$ ) working on the same task as  $X$ , and  $l_x$  is a constant interpreted as *task loyalty*. Thus,  $p(x_B \leftarrow x_A)$  increases with  $|x_B|$  and decreases with  $|x_A|$ . For obvious reasons, upon switching to  $B$ 's task, agent  $A$  will also have to deterministically switch to  $B$ 's home location.

From Eq. 4 and Eq. 6 one can try to predict the dynamics of the reassignment of home locations and tasks. First of all, if there are originally  $N$  agents, each assigned to a unique task and a unique home position, then no conversions will occur, since  $|h_B|$  and  $|x_B|$  in those equations will always be zero. If, however, more than one agent are assigned to one task, then they will eventually "recruit" all other agents and converge to a single homogeneous cluster. If there are several groups of size more than 1 assigned to different tasks, all agents will still end up with the same task and the same home position as time continues indefinitely. We refer to the behavior of such agents as *conformity*, because it resembles the similar phenomenon in human society. The conformist behavior in our model can briefly be summarized as follows: "always follow the majority, both in terms of task and home location".

The idea behind conformity is *cooperation*. Agents should not pursue their own individual goals, which may interfere with each other; rather, they should work as a team. In this case, the team is an example of self-organization; the decision of where and what kind of clusters should be formed is collective and emergent. As experimental results show, conformity helps avoid conflicting goals, e.g., different home locations of the same object type. Conformity also helps improve the clustering performance of the objects of a given type  $x$ , because the number of agents involved in the process of clustering tends to increase up to  $N$ .

It is clear, however, that once agents have all converged to a single task, objects of other types will never become subject to sorting again. Thus, there must be a probability  $p_n$  for a conformist agent to give up its current task and home location and to switch its attention to objects of neglected types. This probability may be fixed. However, we suggest that it may be more reasonable to calculate it by estimating the *performance*  $u$  of the agent's work, e.g., how many useful actions the agent has accomplished within the last  $T_U$  steps. The only actions considered useful are picking up an object or putting it down. Random roaming in search for objects, direct routing to the home location, or random roaming in

search for empty cells to deposit an object are not considered useful actions. We could estimate the performance as follows:

$$u = \min \left( 1, \frac{n_u}{N_U} \right) \quad (7)$$

where  $n_u$  is the number of useful actions accomplished within the last  $T_U$  time steps, and  $N_U$  is the required maximum number of useful actions, at which  $u$  is saturated at 1. For example, we may assume that accomplishing  $N_U = 10$  useful actions within the last  $T_U = 500$  time steps should be considered ideal performance. Note that large values of  $n_u$  with respect to  $T_U$  will hardly be ever achieved, because generally agents spend much more time roaming randomly than picking up or putting down objects.

Having estimated  $u$ , we can calculate  $p_n$  using the following equation:

$$p_n = \left( \frac{1 - u}{1 - u + c} \right)^2 \quad (8)$$

where  $c$  is a constant which we refer to as *conformity threshold*. Thus,  $p_n$  decreases with  $u$  from  $\left( \frac{1}{1 + c} \right)^2$  (when  $u = 0$ ) to 0 (when  $u = 1$ ).

If agent  $A$  has decided to give up its current task  $x_A$  at its current home location  $h_A$ , it starts random walk for a period of time (in our experiments, this period is equal to 300 iterations). Once 300 iterations have passed, the current location of  $A$  becomes its new home location. The behavior of  $A$  is then completely identical to the behavior of the agent that has just started the simulation: that is,  $A$  starts random walk, and the first object met defines  $A$ 's new task  $x'_A$ .  $A$  then starts looking for  $x'$ -objects and brings them to  $h'_A$ .

Suppose that  $x'_A \equiv x_A$  and  $|h_A|$  is relatively large. Then it is likely that  $A$  will sooner or later encounter one of the agents still working at  $h_A$ . Since  $|h_A| > |h'_A|$ ,  $A$  will be likely to convert back to  $h_A$ . This mechanism prevents agents from starting a new cluster of the same type to which the group has previously converged to.

Suppose, however, that  $x'_A \neq x_A$ , and all agents but  $A$  are working on  $x_A$ . Then  $A$  upon encountering any of its former colleagues will be likely to convert back to  $x_A$  (because  $|x_A| > |x'_A|$ ). To prevent this, we consider  $A$  a *nonconformist*. In our model, a nonconformist, i.e., an agent that has recently given up the task carried out by the majority and started executing a new task, gains a special "ability" that allows it not only to keep executing its new task, but also to recruit other agents.

To distinguish formally  $A$  from other agents that are working within groups, we define a measure of *nonconformity*  $\psi$  associated with each agent.  $\psi_A = 1$ , as a reflection of the fact that  $A$  does not conform to the majority. For any agent  $X$  that works within a group,  $\psi_X = 0$ . We now update Eq. 6 taking into consideration nonconformity  $\psi$ :

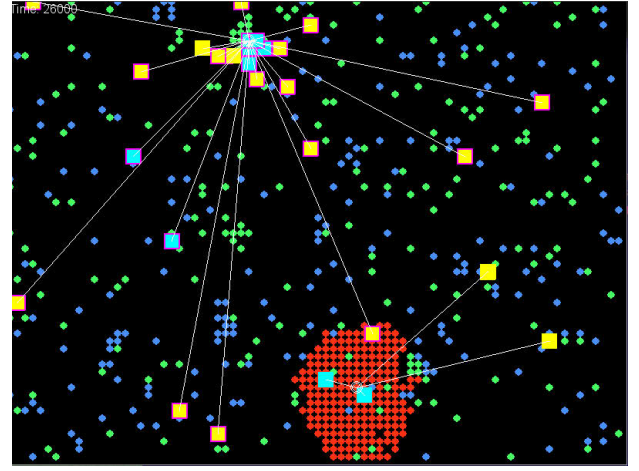


Figure 1: A screenshot from the simulation taken shortly after a nonconformist appeared. Agents are depicted as squares, and objects to be sorted are drawn as smaller circles. There are three types of objects – red, blue, and green pucks. Yellow squares correspond to unladen agents, and blue squares denote agents that are carrying pucks. A magenta border around an agent  $X$  indicates that  $\psi_X > 0$ . For each agent, a white line is drawn to show the distance from its home location. In this experiment, there are 600 pucks (200 pucks of each color),  $N = 30$ , and the grid size is  $80 \times 60$ .

$$p(x_B \leftarrow x_A) = \begin{cases} \min \left[ 1, \frac{\psi_B}{\psi_A + l_c} \right], & \text{if } \psi_A > 0 \text{ or } \psi_B > 0 \\ \min \left[ 1, \left( \frac{|x_B|}{|x_A| + l_c} \right)^2 \right], & \text{if } \psi_A = \psi_B = 0 \end{cases} \quad (9)$$

where  $l_c$  is a constant which can be referred to as *loyalty to crowd*.

Thus, any agent  $A$  with positive nonconformity will have a zero probability of joining a conformist  $B$  (since  $\psi_B = 0$ ). Conformist  $B$ , however, will have a positive probability of joining nonconformist  $A$ :  $p(x_A \leftarrow x_B) = \frac{\psi_A}{l_c} > 0$ . If both  $A$  and  $B$  are conformists, Eq. 9 is reduced to Eq. 6.

The last detail is updating  $\psi$ . For conformists,  $\psi$  always remains zero. Once a conformist  $B$  has been recruited by a nonconformist  $A$ , it accepts its value of nonconformity:  $\psi_B = \psi_A$ . Furthermore, each nonconformist  $A$  decreases its nonconformity as the number of its colleagues increases:

$$\psi'_A = \begin{cases} 0, & \text{if } \psi_A = 0 \\ 1 - \min \left( 1, \frac{|h_A|}{N_G} \right), & \text{otherwise} \end{cases} \quad (10)$$

where  $N_G$  is the maximum size of a group that can be recruited by a nonconformist. Upon reaching this limit,  $A$

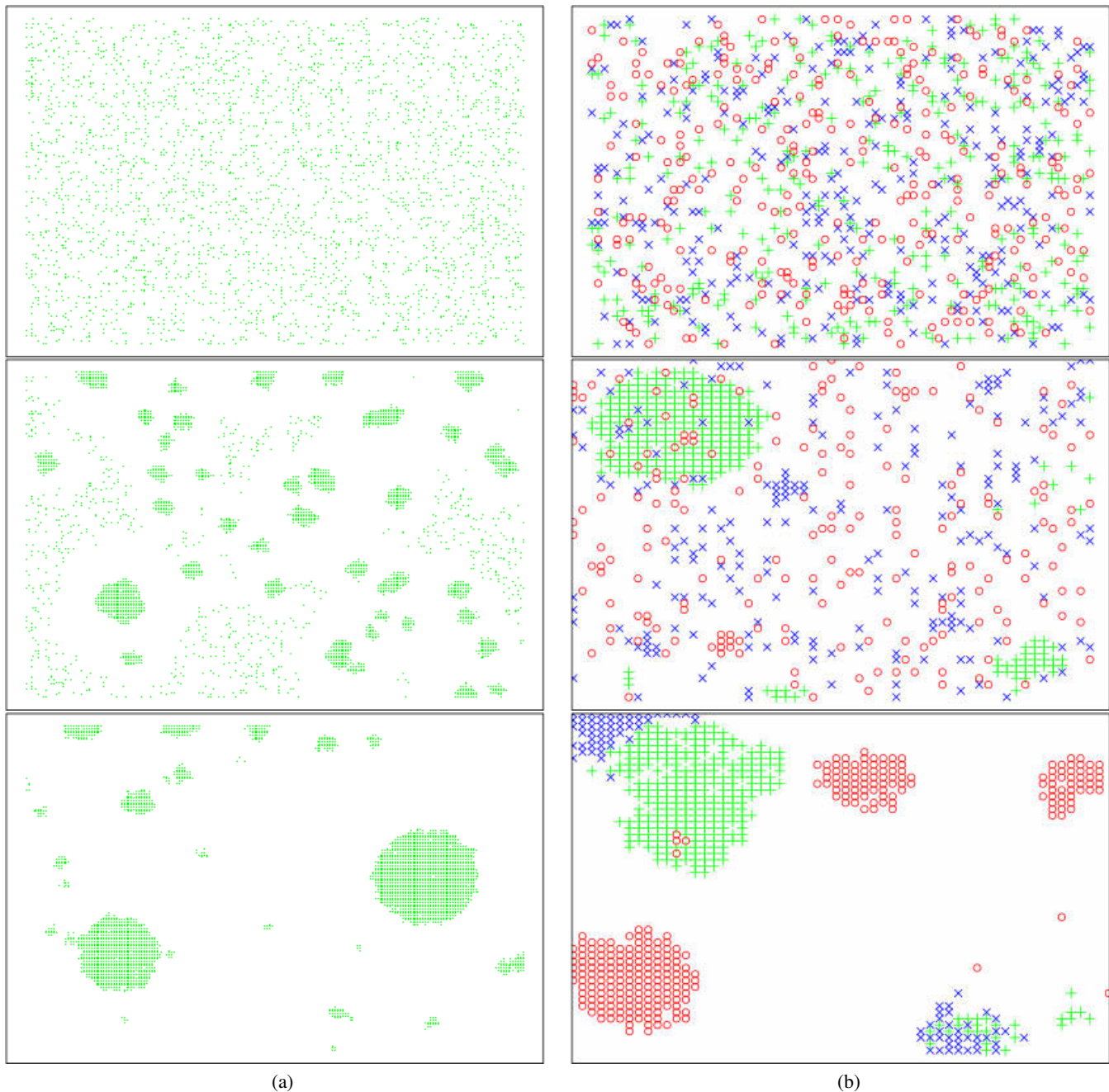


Figure 2: **(a)**: Clustering of 3,000 objects of the same type after 1, 10,000, and 250,000 steps; grid  $200 \times 150$ . The size of the environment is 3-4 times larger than  $r = 50$ ; two stable teams emerge, working at a significant distance from each other. Nonconformists tend to rejoin one of those teams shortly. **(b)**: Sorting of 900 objects of 3 types (300 objects of each type) after 1, 10,000 and 50,000 steps; grid  $80 \times 60$ . The relative magnitudes of the grid size and  $r$  allow the entire agent population work as a single team most of the time. First, a cluster of one type of objects is consistently formed. Then nonconformists start to appear, switching the entire group to new tasks. (Note that pucks being carried by agents are not displayed here.)



ceases its nonconformist status and starts acting like any other conformist agent. Note that  $A$  will not gain nonconformity again from its colleagues which may still be nonconformists (because of using slightly outdated estimations of  $|h_A|$ ). Therefore, the entire group will eventually lose its nonconformist status and will become subject to honest conformist competition with other groups. If  $N_G > \frac{N}{2}$ , then, whenever a nonconformist is spawned, it will tend to create a new majority, switching the entire agent population to a new task.

As an illustration, Fig. 1 shows how the simulation of the sorting process looks like roughly one hundred iterations after a nonconformist appeared. A few important things can be noted. First, just before a nonconformist emerged, the entire population had been collecting red pucks to the cluster situated at the bottom. Since there are very few red pucks left to be collected (in this example, no isolated pucks are left, but often this is not the case), the agents have very low estimations of  $u$ , resulting in  $p_n$  large enough to create a nonconformist. Next, note that the nonconformist has already recruited many other agents which now have  $\psi > 0$ . According to Eq. 9, it is very likely that 4 agents that are still working on red pucks will shortly join the nonconformists. Interestingly enough, a couple of agents working in the “nonconformist” group have already stopped being nonconformists ( $\psi$  has become zero). This is because their  $|h_X|$  values have surpassed the  $N_G$  threshold (see Eq. 10). Agents that are farther away from the nonconformist home location have less chance to communicate with their colleagues; thus, their  $|h_X|$  increase more slowly, and they still consider themselves nonconformists. In 100-200 iterations, the entire population switches to green pucks, and all nonconformists become conformists again. Once almost all green pucks are collected, we can expect another nonconformist to appear.

The model described contains a number of parameters that can be tuned to achieve a desired balance between conformity and nonconformity. The next section offers experimental results showing how different values of some of those parameters affect the performance of the developed social model.

## Experiments

Our experiments are conducted in a Monte-Carlo simulation which is functionally as close as possible to the simulation used by Deneubourg et. al. (Deneubourg et al., 1991). We use a grid-based environment and do not allow cells to be occupied by more than one object and one agent at the same time. Each iteration of the simulation, the agents are updated in random order. During the update cycle, each agent communicates with one random member within its communication range  $r$  and then performs an action depending on its current state and perception (e.g., moves one cell toward its home location, or picks up an object). After the com-

munication session is completed, the agent is not allowed to communicate for the next  $T_S = 5$  time steps (it continues, however, its sorting work). This is done to reduce computational costs of the simulation. In addition, whenever a conformist becomes a nonconformist, it is not allowed to communicate until it settles down at its newly generated position. This is because the new nonconformist does not yet know its own task – it will be determined based on the type of the first object it encounters after its new home location is found. Finally, if an agent tries to become a nonconformist by generating a random number and comparing it with the probability  $p_n$  and fails, its next chance to do so is scheduled in  $T_N = \frac{T_U}{2}$  time steps. This is done to let  $n_u$  accumulate updates before firing  $p_n$  again. The general process of clustering and sorting can be seen in Fig. 2.

To assess the performance of the developed model, we collect two types of statistics: the size of the largest cluster and the number of clusters. We define clusters as follows: 1) An isolated object is a cluster of size 1; 2) An object  $q$  belongs to a cluster  $Q$  if  $q$  is of the same type as objects in  $Q$  and is located in an adjacent cell to any of the objects in  $Q$ .

Each experiment uses the following defaults:  $N_U = 10$ ,  $T_U = 500$ ,  $N_G = \frac{N}{4}$ ,  $N = 50$ ,  $r = 40$ ,  $l_h = l_x = 1$ ,  $l_c = 0.1$ ,  $c = 5$ , grid:  $80 \times 60$ , 3 types of objects, 300 objects of each type. If for any of these parameters another value is used, it is explicitly stated so in the caption of the relevant figure. For each experiment, we ran 30 trials and averaged the results. All plots show mean values accompanied by error bars of  $\pm 1.96$  standard errors of the mean; thus, the error bars correspond to 95% confidence intervals of the mean. For each set of experiments, Shapiro-Wilk tests have been conducted which consistently produced  $p$ -values that are greater than 0.05; thus, there is no reason to reject the hypothesis that our experimental data are not normally distributed.

As it is clear from Fig. 3, the group of socially intelligent agents demonstrates better performance than ALR agents. We included homing agents without a social model into benchmarks in Fig. 3 as well, in order to understand what part of the performance boost acquired by social agents is actually due to their social abilities as opposed to the performance boost gained due to the homing mechanism.

To estimate the influence of different parameters on the overall performance of the model, we conducted a series of experiments where we varied  $l_x$ ,  $c$ ,  $l_h$ ,  $l_c$ , and  $r$ .

As one might expect, smaller values of  $l_x$  result in a relatively quick convergence of the population to one task; thus, the number of clusters decreases relatively slow (because objects of the other two types are consistently ignored), but the size of the largest cluster grows faster (Fig. 4). Note that at  $t \approx 12,000$  the first nonconformists appear; conformists are recruited by nonconformists (see Eq. 9) and start form-



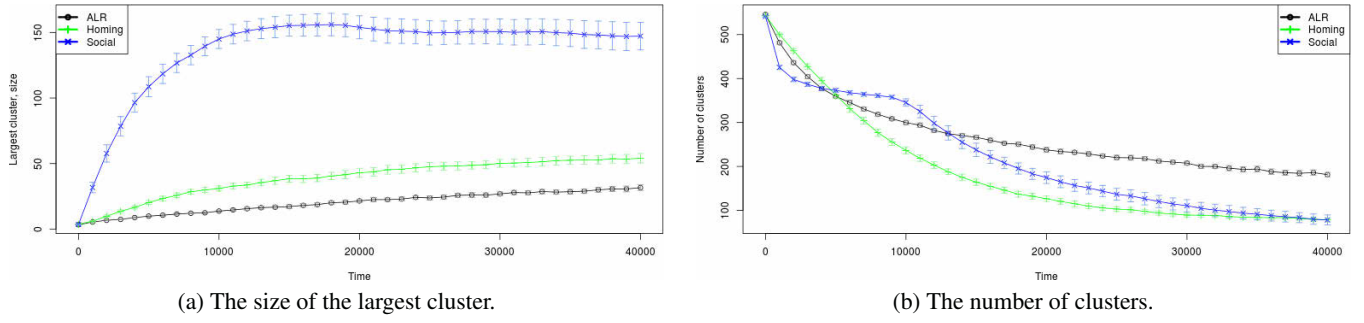


Figure 3: Experimental results for three modes: ALR agents, homing agents (no social model), and social agents; grid:  $120 \times 100$ .

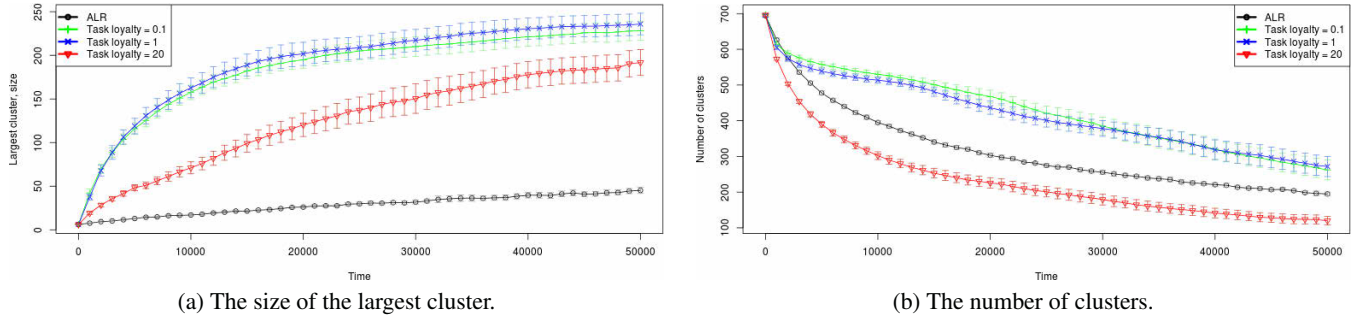


Figure 4: The influence of task loyalty  $l_x$  on the performance. Four modes: ALR agents, social agents with  $l_x = 0.1$ ,  $l_x = 1$ , and  $l_x = 20$ .

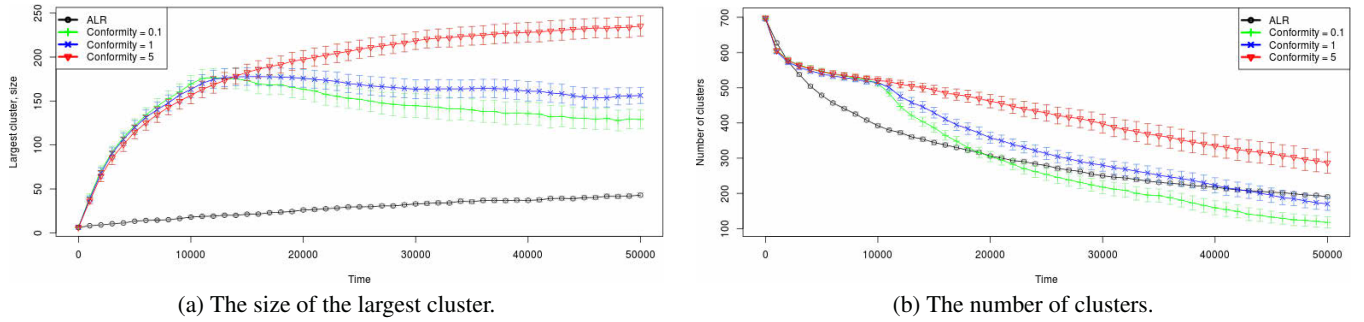


Figure 5: The influence of conformity  $c$  on the performance. Four modes: ALR agents, social agents with  $c = 0.1$ ,  $c = 1.0$ , and  $c = 5.0$ .

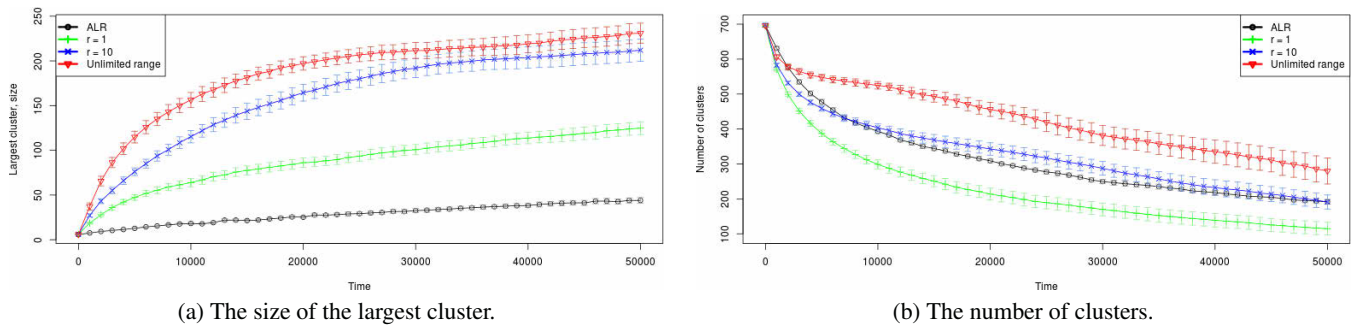


Figure 6: The influence of communication range  $r$  on the performance. Four modes: ALR agents, social agents with  $r = 1$ ,  $r = 10$ , and  $r = \infty$ . The grid size, similarly to previous experiments, is  $80 \times 60$ .

ing larger groups, and, as a result, the number of clusters starts decreasing faster.

Conformity  $c$  prevents agents from defecting from the course of majority. Fig. 5 shows that with large values of conformity the entire population works steadily on one task as few or no nonconformists emerge. Thus, the size of the largest cluster increases, but the overall number of clusters declines relatively slowly. On the contrary, smaller values result in a situation when many nonconformists with different home locations appear, effectively dismantling the agent population. In this case, the largest cluster is gradually destroyed by competing nonconformists.

The influence of the  $l_c$  and  $l_h$  parameters is less significant; relevant figures are omitted due to lack of space.

In our final series of experiments, we estimated the influence of the communication range  $r$  (Fig. 6). Agents with limited  $r$  have less chance to encounter each other; therefore, establishing cooperation is rather unlikely. It is quite natural that such agents show performance similar to non-social homing agents.

Similarly to the trade-off between exploitation and exploration in genetic algorithms, there is a trade-off between conformity and nonconformity in the proposed social model. Conformity is vital for cooperation, avoiding conflicting goals, and convergence; nonconformity, on the other hand, is useful for exploring the task space in search for new goals. Small values of task loyalty  $l_x$  combined with large values of conformity  $c$  may be used to generate agents that tend to work collaboratively and consistently on creating one large cluster, ignoring other tasks. If it is more important to quickly decrease the number of clusters, large values of  $l_x$  and home loyalty  $l_h$  may be used. Note that the described social model can be reduced to the homing algorithm described on page 2 by assuming  $l_x = \infty$ ,  $l_h = \infty$ , and  $c = \infty$ .

## Conclusions

In this paper, we have proposed a model of conformity and nonconformity, a social phenomenon observed in human society. We tackled a well-known problem, collective distributed sorting. Our approach originated in the domain of swarm intelligence, but evolved into *socially intelligent* approach as the social awareness of agents increased. We provided evidence that using the ideas of conformity and nonconformity can be beneficial in artificial multi-agent systems and can increase performance of a task.

The algorithm described in this paper is extreme in the sense that our agents act like *zero-intelligent* particles (Bentley and Ormerod, 2011); that is, the decision to change the home location and/or the task being carried out is based purely on the number of other agents working at that home location and/or on that task. Our experimental results indicate that even this extreme model, based only on social information, is useful and can be applied in collective robotics. Further research could be conducted to reveal

whether adding the social component of intelligence to the agents that are already capable of making informed decisions can increase the effectiveness of the robot group as a whole.

## Acknowledgements

W.B. acknowledges funding from NSERC under the Discovery Grant Program RGPIN 283304-07 and RGPIN 283304-12.

## References

- Aronson, E. (2007). *The Social Animal*. W. H. Freeman and company, New York.
- Bayindir, L. and Sahin, E. (2007). A review of studies in swarm robotics. *Turkish Journal of Electrical Engineering*, 15:115–147.
- Beckers, R. and Holland, O. (1994). From local actions to global tasks: Stigmergy and collective robotics. *Artificial life*, IV:181–189.
- Bentley, A. and Ormerod, P. (2011). Agents, intelligence, and social atoms. In Slingerland, E. and Collard, M., editors, *Creating Consilience: Integrating the Sciences and the Humanities*. Oxford University Press.
- Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.
- Deneubourg, J. L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chretien, L. (1991). The dynamics of collective sorting robot-like ants and ant-like robots. *Simulation of Adaptive Behaviour*, pages 356–363.
- Grech, R., Florez-Revuelta, F., Monekosso, D. N., and Remagnino, P. (2012). Robot teams: Sharing visual memories. In *Distributed Autonomous Robotic Systems*.
- Melhuish, C., H. O. and Hoddell, S. (1998). Collective sorting and segregation in robots with minimal sensing. In *Proceedings of the fifth international conference on simulation of adaptive behavior on From animals to animats*.
- Sahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. *Swarm robotics*, 3342:10–20.
- Seeley, T. D. (2010). *Honeybee democracy*. Princeton University Press.
- Verret, S., Zhang, H., and Meng, M. (2004). Collective sorting with local communication. *Intelligent Robots and Systems*, 3:2687–2692.
- Vorobyev, G., Vardy, A., and Banzhaf, W. (2012). Supervised learning in robotic swarms: From training samples to emergent behavior. In *Distributed Autonomous Robotic Systems*.
- Wang, T. and Zhang, H. (2004). Collective sorting with multiple robots. *Robotics and Biomimetics*, pages 716–720.

# A Neuromechanical Controller of a Hexapod Robot for Walking on Sponge, Gravel and Snow Surfaces

Xiaofeng Xiong<sup>1</sup>, Florentin Wörgötter<sup>1</sup> and Poramate Manoonpong<sup>1</sup>

<sup>1</sup>Bernstein Center for Computational Neuroscience (BCCN), The Third Institute of Physics,  
Georg-August-Universität Göttingen, D-37077 Göttingen, Germany  
(xiong, worgott, poramate)@physik3.gwdg.de

## Abstract

Physiological studies suggest that the integration of neural circuits and biomechanics (e.g., muscles) is a key for animals to achieve robust and efficient locomotion over challenging surfaces. Inspired by these studies, we present a neuromechanical controller of a hexapod robot for walking on soft elastic and loose surfaces. It consists of a modular neural network (MNN) and virtual agonist-antagonist mechanisms (VAAM, i.e., a muscle model). The MNN coordinates 18 joints and generates basic locomotion while variable joint compliance for walking on different surfaces is achieved by the VAAM. The changeable compliance of each joint does not depend on physical compliant mechanisms or joint torque sensing. Instead, the compliance is altered by two internal parameters of the VAAM. The performance of the controller is tested on a physical hexapod robot for walking on soft elastic (e.g., sponge) and loose (e.g., gravel and snow) surfaces. The experimental results show that the controller enables the hexapod robot to achieve variably compliant leg behaviors, thereby leading to more energy-efficient locomotion on different surfaces. In addition, a finding of the experiments complies with the finding of physiological experiments on cockroach locomotion on soft elastic surfaces.

## Introduction

There are increasing demands for robots to walk on a series of diverse terrains (Ozcan et al., 2010; Qian et al., 2012). However, few robots can walk on soft elastic (e.g., sponge) and loose (e.g., gravel and snow) surfaces. This is because traversing these surfaces efficiently requires variable compliance of legs (Spence, 2011; Bermudez et al., 2012). Traditionally, the variable compliance of legged robots can be achieved by passive compliance mechanisms (Ham et al., 2009) and/or active compliance control (Görner and Hirzinger, 2010). For example, by using active compliance control with joint torque feedback, a hydraulically actuated quadruped robot (i.e., HyQ, 90 kg) has been developed for moving over terrains (Boaventura et al., 2012). Nevertheless, the complex mechanical and sensing components of the HyQ robot greatly increase its size and mass, thereby not fitting for developing small legged robots. Yet a small six-legged robot (i.e., EduBot, 3 kg) has been designed by using physically passive variable compliant legs

(Galloway et al., 2011). The experimental results show that stiffer legs allow its faster locomotion on soft surfaces.

In contrast to the robot experimental results, owing to energy efficiency, biological study has shown that cockroaches (i.e., *Blaberus discoidalis*) use their softer legs on soft surfaces (Spence et al., 2010; Spence, 2011). This finding reveals a neuromechanical control strategy of hexapod locomotion on soft surfaces. In fact, the strategy is not the result of a single component rather interactions between a nervous system, a musculoskeletal system and the environment. Inspired by this, the work here proposes a novel neuromechanical controller of a hexapod robot for walking on soft elastic and loose surfaces. The neuromechanical controller consists of a modular neural network (MNN) coordinating leg movement and virtual agonist-antagonist mechanisms (VAAM) changing the compliance of legs. The changeable compliance is simply achieved by altering two internal parameters of the VAAM without physical passive compliant mechanisms (Ham et al., 2009) or joint torque sensing (Görner and Hirzinger, 2010). Employing this controller allows the robot to walk on different surfaces with energy efficiency. Besides, a finding of robot walking complies with the finding of physiological experiments on cockroach locomotion on soft elastic surfaces (Spence et al., 2010; Spence, 2011).

## Neuromechanical Controller of a Hexapod Robot

The experimental robot is a hexapod robot (5.4 kg) (see Fig. 1 (a)). Each three-jointed leg has a TC (Thoraco Coxal) joint allowing the motions of forward and backward, a CTr (Coxa Trochanteral) joint allowing the motions of elevation and depression, and a FTi (Femur Tibia) joint allowing the motions of extension and flexion (see Fig. 1 (b)). Each joint is physically driven by a standard servo motor. There is a force sensor used for detecting the analog signal at each leg (see  $f_{c1-6}$  in Fig. 1 (a)). A current sensor installed inside the body of the hexapod robot is used to detect electrical current used for all motors and sensors of the hexapod robot. For more details of the hexapod robot, we refer to (Manoonpong et al., 2013).

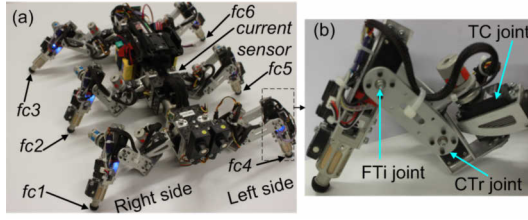


Figure 1: A hexapod robot (a) Six legs and six foot sensors  $fc_{(1-6)}$ . (b) A three-jointed leg.

## Modular Neural Network (MNN)

The modular neural network (MNN) is a biologically-inspired hierarchical neural controller (McCrea and Rybak, 2008), which generates signals for leg and joint coordination of the hexapod robot. The MNN consists of a central pattern generator (CPG, see Fig. 2 (a)), a phase switch module (PSM, see Fig. 2 (b)) and two velocity regulating modules (VRMs, see Fig. 2 (c)). All neurons of the MNN are modelled as discrete-time non-spiking neurons. The activity  $H_i$  of each neuron develops according to:

$$H_i(t) = \sum_{j=1}^m W_{ij} o_j(t-1) + B_i, i = 1, \dots, m, \quad (1)$$

where  $m$  denotes the number of units,  $B_i$  is an internal bias term (i.e., stationary input) to neuron  $i$ ,  $W_{ij}$  is the synaptic strength of the connection from neuron  $j$  to neuron  $i$ . The output  $o_i$  of all neurons of the MNN is calculated using a hyperbolic tangent ( $\tanh$ ) transfer function, i.e.,  $o_i = \tanh(H_i), \in [-1, 1]$ . The CPG consists of only two neurons with full connectivity (see Fig. 2(a)), where  $B_1 = B_2 = 0.01$ . The weights  $W_{12}$  and  $W_{21}$  are given by:

$$W_{12}(S) = 0.18 + S, W_{21}(S) = -0.18 - S, \quad (2)$$

where  $S \in \mathbb{R}_{[0,0.18]}$  is the input of the modular neural network, which determines walking patterns of the hexapod robot. The speed of its leg motion increases with increasing  $S$ . Here, we set  $S = 0.04$  resulting in slow walking behavior, which leads to stable and energy-efficient locomotion on non-flat surfaces (Manoonpong et al., 2013).

The PSM is a generic feed-forward network consisting of three hierarchical layers with ten hidden neurons (i.e.,  $H_3 - H_{12}$ ). The outputs of the PSM are projected to the FTi (i.e.,  $F(R, L)_{(1,2,3)}$ ) and CTr (i.e.,  $C(R, L)_{(1,2,3)}$ ) motor neurons (see Fig. 2 (d)), as well as the neurons  $H_{13}$  and  $H_{14}$  of the two velocity regulating modules (VRMs, see Fig. 2 (c)). The two VRMs are feed-forward networks projecting their outputs to the TC motor neurons  $T(R, L)_{(1,2,3)}$  (see Fig. 2 (d)). In the neuromechanical controller, the outputs  $N_{1-18}$  of the motor neurons are the neural activations of 18

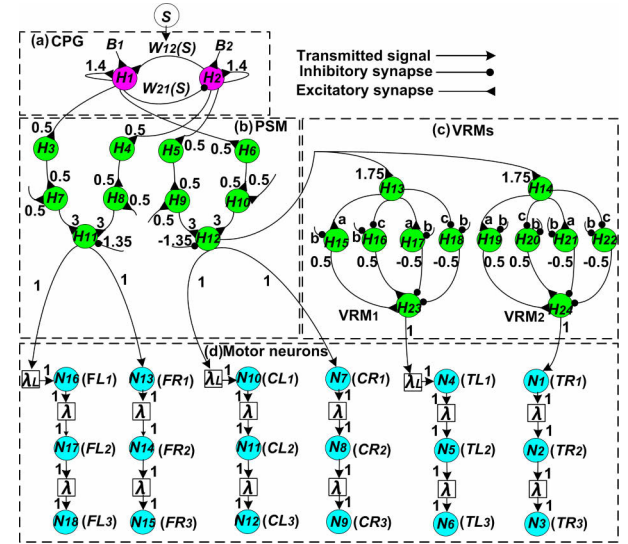


Figure 2: Modular neural network. Output neurons (i.e.,  $N_{(1-18)}$ ) represent the neural activations of 18 joints of the hexapod robot. All connection strengths together with bias terms are indicated by the small numbers except some parameters of the VRMs ( $a = 1.7246, b = -2.48285, c = -1.7246$ ). Delays  $\lambda_L$  and  $\lambda$  between motor neurons (i.e.,  $N_{1-18}$ ) are set to:  $\lambda_L = 48$  time steps,  $\lambda = 16$  time steps. Abbreviation are:  $TR(L)_{1,2,3}$  = TC joints of the Right(Left) Front, Middle, Hind legs,  $CR(L)_{1,2,3}$  = CTr joints of the Right(Left) Front, Middle, Hind legs,  $FR(L)_{1,2,3}$  = FTi joints of the Right(Left) Front, Middle, Hind legs.

joints of the hexapod robot.  $N_{1-18}$  enable its legs to perform fast swing and slow stance phases (see Fig. 3). Delays  $\lambda_L$  and  $\lambda$  between the outputs of motor neurons are fixed (see Fig. 2 (d)). For more details of the MNN, we refer to our previous work (Manoonpong et al., 2013). However, the previous work did not consist of muscle-like mechanisms (e.g., virtual agonist-antagonist mechanism (VAAM)). Including the VAAM allows the hexapod robot to achieve more energy-efficient locomotion (described below).

## Virtual Agonist-antagonist Mechanism (VAAM)

The virtual agonist-antagonist mechanism (VAAM) consists of a pair of agonist and antagonist mechanisms (see Fig. 4(a)). They produce active and passive forces by its contractile and parallel elements (CEs and PEs, see Fig. 4(b)). In Fig. 4(a), the physical joint is driven by a pair of the virtual agonist-antagonist mechanism (VAAM, i.e.,  $M1$  and  $M2$ ). 'Virtual' means that the physical joint, physically driven by a standard servo motor, imitates muscle-like behaviors as if it were driven by a pair of agonist and antagonist muscles. The joint actuation relies on the CEs while the PEs govern joint compliance.

The parallel elements (i.e., PEs) are modelled as spring-



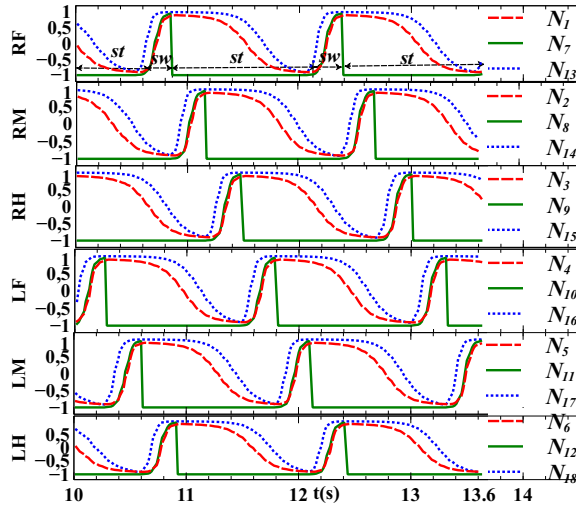


Figure 3: Outputs of motor neurons  $N_{1-18}$  between 10 s and 13.6 s. Abbreviations are: R(F,M,H) = Right (front, middle, hind) legs, L(F,M,H) = Left (front, middle, hind) legs. *st* = stance phase, *sw* = swing phase.

damper systems (see Fig. 4(b)). The matrix  $[f_1^P, f_2^P]^T$  of passive forces created by  $PE_{(1,2)}$  is the sum of two Hadamard products:

$$[f_1^P, f_2^P]^T = \Gamma_{2 \times 1} \circ L_{2 \times 1} + \Phi_{2 \times 1} \circ V_{2 \times 1}, \quad (3)$$

where

- $\Gamma_{2 \times 1}$  is the matrix of stiffness coefficients of  $PE_{(1,2)}$ , i.e.,  $\Gamma_{2 \times 1} = [K, K]^T$ ;
- $L_{2 \times 1}$  is the matrix of displacements of  $PE_{(1,2)}$ , i.e.,  $L_{2 \times 1} = [l_1^P - l_0, l_2^P - l_0]^T$ .  $l_0$  is the initial length of  $PE_{(1,2)}$ , which is set to:  $l_0 = 0.085$ ;
- $\Phi_{2 \times 1}$  is the matrix of damper coefficients of  $PE_{(1,2)}$ , i.e.,  $\Phi_{2 \times 1} = [D, D]^T$ ;
- $V_{2 \times 1}$  is the matrix of velocities of  $PE_{(1,2)}$ , i.e.,  $V_{2 \times 1} = [v_1^P, v_2^P]^T$ .

The active forces produced by the CEs are approximated by the product of the neural activation  $N_j$  and the activation intensities  $i_{(1,2)}$ . The matrix  $[f_1^C, f_2^C]^T$  of active forces generated by  $CE_{(1,2)}$  are represented by (see Fig. 4 (b)) :

$$[f_1^C, f_2^C]^T = N_j \times [i_1, i_2]^T, \quad (4)$$

where

- $N_j$  is the neural activation of  $CE_{(1,2)}$  (i.e.,  $N_j \in [-1, 1]$ ). It is one of the outputs  $N_{1-18}$  of the MNN (see Fig. 2 (d));
- $[i_1, i_2]^T$  is the matrix of activation intensities for  $CE_{(1,2)}$  (i.e.,  $i_{(1,2)} \in [-1, 1]$ ).

The total forces  $f_1^T$  and  $f_2^T$  are the sum of the active and passive forces produced by  $M1$  and  $M2$ . They are given by (derived from Eqs.(3) and (4)):

$$f_1^T = f_1^P + f_1^C = \underbrace{K(l_1^P - l_0) + Dv_1^P}_{f_1^P} + \underbrace{N_j i_1}_{f_1^C}, \quad (5)$$

$$f_2^T = f_2^P + f_2^C = \underbrace{K(l_2^P - l_0) + Dv_2^P}_{f_2^P} + \underbrace{N_j i_2}_{f_2^C}. \quad (6)$$

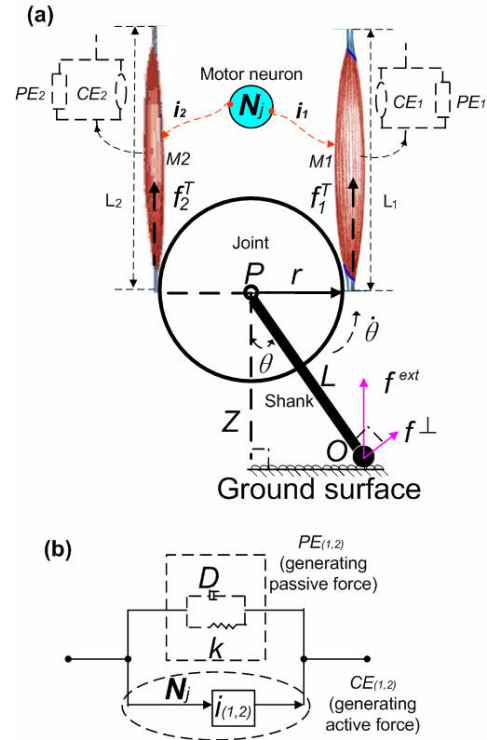


Figure 4: Virtual agonist-antagonist mechanism (VAAM) for joint control interacting with the ground surface. (a) A physical joint is driven by a pair of the VAAM with the lengths  $L_1$  and  $L_2$  (i.e.,  $M1$  and  $M2$ ). The interaction results in the force  $f^{ext}$ , which drives the joint  $P$  with the radius  $r$  via the shank with the length  $L$ .  $f^{ext}$  is sensed by a force sensor (i.e.,  $O$ ), and  $f^\perp$  is the amount of  $f^{ext}$  directed perpendicularly to the position of the joint  $P$ . (b) The VAAM consists of contractile (i.e.,  $CE_{(1,2)}$ ) and parallel (i.e.,  $PE_{(1,2)}$ ) elements for producing active and passive forces.

The antagonist mechanism  $M2$  (see Fig. 4 (a)) resists the extension of the joint angle  $\theta$  when receiving the force  $f^{ext}$ , which is sensed by a force sensor. Simultaneously, the agonist mechanism  $M1$  (see Fig. 4 (a)) produces opposing force against  $M2$ . Therefore, the directions of  $f_1^T$  and  $f^{ext}$  are counter-clockwise when the direction of  $f_2^T$  is clockwise.

Their torques acting on the joint  $P$  (see Fig. 4 (a)) are represented by:

$$\tau(f_1^T) = f_1^T r = (K(l_1^P - l_0) + Dv_1^P + N_j i_1) r, \quad (7)$$

$$\tau(f_2^T) = -f_2^T r = -(K(l_2^P - l_0) + Dv_2^P + N_j i_2) r, \quad (8)$$

$$\tau(f^{ext}) = f^\perp L = f^{ext} \sin(\theta) L, \quad (9)$$

where  $r$  is the radius of the joint  $P$ .  $f^\perp$  is the amount of  $f^{ext}$  directed perpendicularly to the position of the joint  $P$ .  $L$  is the length of the shank of the joint  $P$ . Note that the direction of torque  $\tau(f_2^T)$  is opposite to those of  $\tau(f_1^T)$  and  $\tau(f^{ext})$ . We consider the torque pointing outward from the page as the positive torque (e.g.,  $\tau(f_1^T)$  and  $\tau(f^{ext})$ ).

We apply Euler's laws of motion to the rotation of the joint  $P$  (see Fig. 4(a)). The net torque  $\sum \tau$  acting on the joint  $P$  is equal to the product of its moment of inertia  $I$  and angular acceleration  $\ddot{\theta}$ . It is given by:

$$I\ddot{\theta} = \sum \tau = \tau(f^{ext}) + \tau(f_1^T) + \tau(f_2^T). \quad (10)$$

Derived by Eq.(10) (see details in Appendix A), the motion equation of the joint  $P$  is given by:

$$I\ddot{\theta} = \underbrace{f^{ext} \sin(\theta) L}_{\text{torque by } f^{ext}} + \underbrace{r N_j}_{\text{torque by } f_{(1,2)}^C} - \underbrace{r(2K\theta r + 2D\dot{\theta} r)}_{\text{torque by } f_{(1,2)}^P}. \quad (11)$$

Equation (11) governs  $\theta$  of the joint  $P$  driven by the VAAM that is activated by the output  $N_j$  ( $j \in \mathbb{Z}_{[1,18]}$ ) of the MNN.

### Neuromechanical Control Strategies for a Hexapod Robot

The outputs  $O_{1-18} \in \mathbb{R}_{[-1,1]}$  (see Fig.5) of the neuromechanical controller are linearly scaled and transmitted to control the position of the standard servo motors driving the 18 joints of the hexapod robot. Different control strategies are applied in swing and stance phases.

#### Swing phase

When a leg is in swing phase (i.e.,  $f_i^{ext} = 0$ ,  $i \in \mathbb{Z}_{[1,6]}$ , see Fig. 5 (a)), the outputs  $O_{(i,i+6,i+12)}$  of its TC, CTr and FTi joints receive motor neuron signals  $N_{(i,i+6,i+12)}$  of the MNN as their inputs. They satisfy:

$$[O_i, O_{i+6}, O_{i+12}]^T = [0.4N_i, 0.15N_{i+6}, -0.02N_{i+12}]^T - [0.05, 0.86, 0.43]^T, i \in \mathbb{Z}_{[1,6]}. \quad (12)$$

#### Stance phase

Since there is only detection for vertical foot force in the leg, the TC joint allowing only horizontal motions is not effected by a pair of the PEs of the VAAM. Moreover,

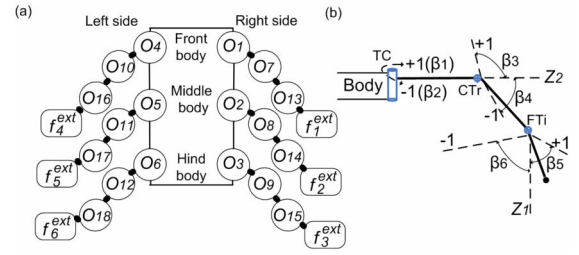


Figure 5: The outputs  $O_{1-18}$  of the neuromechanical controller. (a)  $O_{1-18}$  control the 18 joints of the hexapod robot.  $f_{1-6}^{ext}$  are six analog signals, which are detected by the force sensors at the legs. (b) Relationship between  $O_{1-18}$  and  $\theta_{1-18}$ . The angle ranges of the TC, CTr and FTi joints are as follows:  $[\beta_1, \beta_2] = [0.785, -0.785]rad$ ,  $[\beta_3, \beta_4] = [-1.745, 0.785]rad$ ,  $[\beta_5, \beta_6] = [0.96, -1.222]rad$ .

we test two control setups (see Fig.6) for the FTi joint when the CTr joint is controlled by a pair of the PEs and CEs of the VAAM. The control setups are tested in a physical simulator (i.e., Iprzrobots simulator (Der and Martius, 2012)). The results of the physical simulation show that the FTi joint, purely controlled by a pair of the PEs of the VAAM, allows the hexapod robot to achieve the coordinated movement and stable locomotion (see Figs.6 (a) and (b)). The video clip of the test can be seen at <http://www.youtube.com/watch?v=fMLf6nIOWpM>.

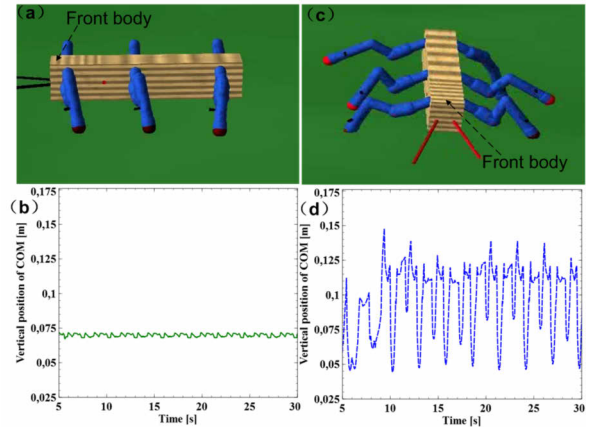


Figure 6: Two control setups for the FTi joint tested in a physical simulator. (a) Snapshot of stable walking of the hexapod robot at 15 s. (b) Vertical position of its body. (c) Snapshot of unstable walking of the hexapod robot at 15 s. (d) Vertical position of its body.

Therefore, the control strategy of its three-jointed legs during stance phase is as follows: each TC joint (i.e., proximal joint) is purely controlled by a pair of the CEs of the VAAM (i.e., pure actuation), each CTr joint (i.e., intermediate joint) is governed by a pair of the CEs and PEs of the

VAAM (the combination of actuation and compliance), and each FTi joint (i.e., distal joint) is driven by a pair of the PEs (i.e.,  $PE_1$  and  $PE_2$ ) of the VAAM (i.e., pure compliance) (see Fig. 7). The control strategy is also comparable to the findings revealed by three-jointed leg locomotion of the BigDog-inspired study (Lee et al., 2008; Raibert, 2008).

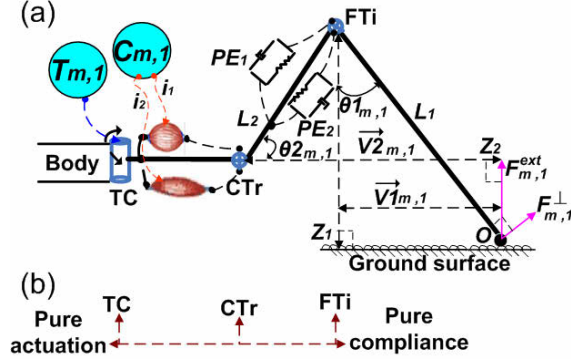


Figure 7: Control framework for a three-jointed leg of the hexapod robot in stance phase. (a) The three-jointed legs take the strategy of directional actuation and compliance (see text for details). (b) The control strategy for the leg. The function of compliance intensifies from the TC to FTi joints.

The relationship between the outputs  $O_{1-18}$  and the angles  $\theta_{1-18}$  of the joints is shown in Fig. 5(b). In concrete, the computations of  $O_{1-18}$  are as follows:

**FTi joints :** Each FTi joint is only driven by  $PE_{(1,2)}$  of the VAAM (see Fig. 7 (a)). Therefore, their neural activations  $N_{6 \times 1}$  are equal to zero,

$$N_{6 \times 1} = [0, 0, \dots, 0]^T, \quad (13)$$

where  $N_{6 \times 1} = [N_{13}, N_{14}, \dots, N_{18}]^T$ .

In addition, the forces  $f_{1-6}^{ext}$  directly result in the extension and flexion of the FTi joints. Therefore, the matrix  $\tau_{6 \times 1}^{FTi}$  of torques acting on the FTi joints is given by (derived from Eq.(9)):

$$\tau_{6 \times 1}^{FTi} = F_{6 \times 1}^\perp L_1, \quad (14)$$

where

$$\begin{aligned} F_{6 \times 1}^\perp &= F_{6 \times 1}^{ext} \circ \sin(\theta_{16 \times 1}) \\ &= [f_1^{ext} \sin(\theta_{13}), f_2^{ext} \sin(\theta_{14}), \dots, f_6^{ext} \sin(\theta_{18})]^T, \end{aligned}$$

Substituting Eqs.(13) and (14) into Eq.(11),  $\theta_{16 \times 1}$  is the sum of the Hadamard products:

$$\begin{aligned} I\ddot{\theta}_{16 \times 1} &= F_{6 \times 1}^{ext} \circ \sin(\theta_{16 \times 1}) L_1 \\ &\quad - r(2rK_{16 \times 1} \circ \theta_{16 \times 1} + \\ &\quad 2rD_{16 \times 1} \circ \dot{\theta}_{16 \times 1}), \end{aligned} \quad (15)$$

where

$$\begin{aligned} \ddot{\theta}_{16 \times 1} &= [\ddot{\theta}_{13}, \dots, \ddot{\theta}_{18}]^T, \dot{\theta}_{16 \times 1} = [\dot{\theta}_{13}, \dots, \dot{\theta}_{18}]^T, \\ K_{16 \times 1} &= [K_{13}, \dots, K_{18}]^T, D_{16 \times 1} = [D_{13}, \dots, D_{18}]^T, \end{aligned}$$

The angles  $\theta_{1m,1}$  ( $m \in \mathbb{Z}_{[1,6]}$ ) of the FTi joints can be linearly transformed into their outputs  $O_j$  (see Fig. 5).  $O_j$  are given by ( $j \in \mathbb{Z}_{[13,18]}$ ):

$$O_j = 0.92\theta_{j-12,1} + 0.12. \quad (16)$$

**CTr joints :** Each CTr joint is driven by  $PE_{(1,2)}$  and  $CE_{(1,2)}$  of the VAAM.  $CE_{(1,2)}$  are activated by one of the outputs  $N_{7-12}$  of the MNN (see Fig. 7 (a)). For example, the pair of the VAAM of the right front CTr joint (i.e., CR1) is activated by  $N_7$  of the MNN (see Fig. 2 (d)). The forces  $f_{1-6}^{ext}$  indirectly result in the elevation and depression of the CTr joint. The matrix of the CTr joint angles is  $\theta_{26 \times 1} = [\theta_7, \theta_8, \dots, \theta_{12}]^T$ . The computation of the torques generated by  $f_{1-6}^{ext}$  needs to be approximated, since there are no torque sensors at the CTr joint. Therefore, the matrix  $\tau_{6 \times 1}^{CTr}$  of the torques acting on the CTr joints is given by:

$$\begin{aligned} \tau_{6 \times 1}^{CTr} &= F_{6 \times 1}^{ext} \circ \vec{V}_{26 \times 1} \\ &= F_{6 \times 1}^{ext} \circ (L_2 \cos(\theta_{26 \times 1}) + \vec{V}_{16 \times 1}), \end{aligned} \quad (17)$$

where

$$\begin{aligned} L_2 \cos(\theta_{26 \times 1}) &= L_2 [\cos(\theta_7), \cos(\theta_8), \dots, \cos(\theta_{12})]^T, \\ \vec{V}_{16 \times 1} &= L_1 [\sin(\theta_{13}), \sin(\theta_{14}), \dots, \sin(\theta_{18})]^T, \end{aligned}$$

Substituting Eq.(17) to Eq.(11), the matrix  $\theta_{26 \times 1}$  of the CTr angles is the sum of the Hadamard products:

$$\begin{aligned} I\ddot{\theta}_{26 \times 1} &= F_{6 \times 1}^{ext} \circ (L_2 \cos(\theta_{26 \times 1}) + \vec{V}_{16 \times 1}) \\ &\quad + [rC_{6 \times 1} - 2r^2(K_{26 \times 1} \circ \theta_{26 \times 1} + \\ &\quad D_{26 \times 1} \circ \dot{\theta}_{26 \times 1})], \end{aligned} \quad (18)$$

where

$$\begin{aligned} \ddot{\theta}_{26 \times 1} &= [\ddot{\theta}_7, \dots, \ddot{\theta}_{12}]^T, \dot{\theta}_{26 \times 1} = [\dot{\theta}_7, \dots, \dot{\theta}_{12}]^T, \\ K_{26 \times 1} &= [K_7, \dots, K_{12}]^T, D_{26 \times 1} = [D_7, \dots, D_{12}]^T, \end{aligned}$$

The angles  $\theta_{2m,1}$  ( $m \in \mathbb{Z}_{[1,6]}$ ) of the CTr joints are linearly transformed into their outputs  $O_j$  (see Fig. 5).  $O_j$  are given by ( $j \in \mathbb{Z}_{[7,12]}$ ):

$$O_j = -0.8\theta_{j-6,1} - 0.38. \quad (19)$$

**TC joints :** All TC joints are purely controlled by  $CE_{(1,2)}$  of the VAAM.  $CE_{(1,2)}$  are activated by the outputs  $N_{1-6}$  of the MNN (see Fig. 7 (a)).  $N_{1-6}$  are linearly transformed into the outputs  $O_{1-6}$  of the TC joints. The matrix of the TC neuron outputs is  $T_{6 \times 1} = [N_1, N_2, \dots, N_6]^T$ .  $O_j$  are given by ( $j \in \mathbb{Z}_{[1,6]}$ ):

$$O_j = 0.4T_{j,1} - 0.05. \quad (20)$$

## Experimental Results

The proposed neuromechanical and pure neural controllers were implemented on the hexapod robot for walking on soft elastic (i.e., sponge) and loose (i.e., gravel and snow) surfaces. Changing the matrices of the stiffness coefficients of the FTi (i.e.,  $K1_{6 \times 1}$  in Eq.(15)) and CTr (i.e.,  $K2_{6 \times 1}$  in Eq.(18)) joints enables the legs of the hexapod robot to show variable compliance (see notations in Appendix B). Note that here, all damper coefficients of the CTr and FTi joints were set to:  $D(1,2)_{6 \times 1} = [0.1, 0.1, \dots, 0.1]^T$ . Due to the damper properties of the VAAMs, the noise of force sensor signals is filtered. Hence, we tested three setups for hexpod walking on the surfaces:

- Neuromechanical controller with high stiffness (HSC).  $K(1,2)_{6 \times 1}$  are set as:  $K1_{6 \times 1} = [4, 4, 6, 4, 4, 6]^T$  and  $K2_{6 \times 1} = [8, 8, \dots, 8]^T$ .
- Neuromechanical controller with low stiffness (LSC).  $K(1,2)_{6 \times 1}$  are set as:  $K1_{6 \times 1} = [3, 3, 5, 3, 3, 5]^T$  and  $K2_{6 \times 1} = [6, 6, \dots, 6]^T$ .
- Pure neural controller (PNC).

The pure neural controller (PNC) uses the outputs of the motor neurons of the MNN to directly drive the 18 joints of the robot. The computations of its outputs follow Eq.(12) for stance and swing phases. The free parameters of the proposed neuromechanical controller are chosen based on trial and error. The parameters of three setups allow the hexapod robot to achieve coordinated and stable locomotion, which have been tested in a physical simulator (i.e., Ipzrobots simulator (Der and Martius, 2012)). For each setup, the runs over each surface were repeated until ten successful runs<sup>1</sup> were obtained. For a successful run, the power consumption  $P_i$  and forward velocity  $v_i$  are given by:

$$P_i = 5A_i, v_i = \frac{\Delta DIS_i}{\Delta t}, i \in \mathbb{Z}_{[1,10]}, \quad (21)$$

where 5 Volts is the input voltage of the electrical board and motors of the hexpod robot.  $A_i$  is an average electrical current measured using a current sensor.  $\Delta DIS_i$  is the forward displacement during a time interval  $\Delta t$ . The performance of the runs was measured by "specific resistance"  $\varepsilon_i$  (Gregorio et al., 1997; Saranli et al., 2001).  $\varepsilon_i$  is determined by power consumption  $P_i$  and forward velocity  $v_i$ :

$$\varepsilon_i = \frac{P_i}{mgv_i} = \frac{P_i}{52.974v_i}, \varepsilon_{avg} = \frac{\sum_{i=1}^{10} \varepsilon_i}{10} \quad (22)$$

where  $mg$  is the weight of the hexapod robot, i.e.,  $mg = 52.974$  N. Lower  $\varepsilon_{avg}$  corresponds to more energy-efficient walking, which is desirable.

<sup>1</sup>The data of unsuccessful runs was discarded. In unsuccessful runs, the hexapod robot walked in unwanted directions.

Table 1: Average specific resistances  $\varepsilon_{avg}$  with standard deviations of the hexapod robot walking on sponge, gravel and snow surfaces

| Setup | Sponge               | Gravel               | Snow               |
|-------|----------------------|----------------------|--------------------|
| HSC   | 21.8 ( $\pm 0.9$ )   | 17.2 ( $\pm 0.7$ )   | 18.8 ( $\pm 0.5$ ) |
| LSC   | 19.7 ( $\pm 0.8$ )   | 29.3 ( $\pm 2.0$ )   | 22.3 ( $\pm 0.8$ ) |
| PNC   | 542.4 ( $\pm 63.8$ ) | 112.7 ( $\pm 13.0$ ) | -                  |

### Sponge surface

The interval  $\Delta t$  over one run was 27 s. A 1.5 m long sponge (i.e., three pieces of sponge glued together) was used as a soft elastic surface. The experiment result is shown in Table 1 and Fig. 8. The hexapod robot that was controlled by the neuromechanical controllers with the low (i.e., LSC) or high (i.e., HSC) stiffness consumed less energy than controlled by the pure neural controller (i.e., PNC). This is because LSC and HSC allow for variable joint compliance of the hexapod robot resulting in leg adaptations to sponge deformations (see Fig. 8 (a)). The experimental video can be seen at the link <http://www.youtube.com/watch?v=vEqylwMXfJE>.

Interestingly, LSC shows the lowest average specific resistance with 19.7. This experimental result shows that softer legs (i.e., LSC setup) allow the hexapod robot to achieve more energy-efficient locomotion, compared to stiffer legs (i.e., HSC setup). The finding complies with a finding of physiological experiments on cockroach locomotion. Owing to energy efficiency, cockroaches (i.e., *Blaberus discoidalis*) also use their softer legs on soft elastic surfaces (Spence et al., 2010; Spence, 2011).

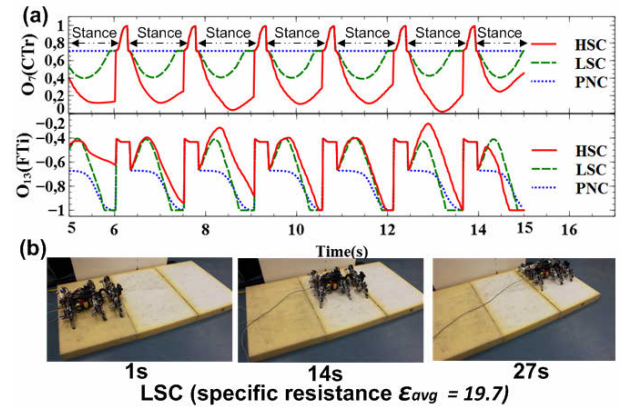


Figure 8: Comparisons of HSC, LSC, and PNC for walking on sponge surface. (a) Control signals  $O_7$  and  $O_{13}$  for the CTr and FTi joints of the right front leg. There are seven stance and six swing phases between 5 s and 15 s. (b) A series of photos shows hexapod robot walking controlled by LSC.



## Gravel and snow surfaces

The interval  $\Delta t$  over one run was 60 s. Gravel surface is a bed (i.e., the length is 2.4 m) of loosely packed gravels (i.e., gravel diameter  $\phi$  : 5 mm - 60 mm). The experimental result is shown in Table 1 and Fig. 9. HSC and LSC enable the hexapod robot to adapt its joint motions to different sizes of gravels (see Joint motion I and II in Fig.9 (a)), while PNC does not adapt the joint motions leading to difficulty of locomotion. In addition, the average specific resistance was lowest for HSC (i.e.,  $\varepsilon_{avg} = 17.2$ ), thereby leading to more energy-efficient locomotion. This is because HSC allows the legs of the hexapod robot to penetrate more deeply into gravel surface (see control signals  $O_7$  in Fig.9 (a)). The experimental video of walking on gravel surface can be seen at <http://www.youtube.com/watch?v=f2G4UzUQ6Iw>.

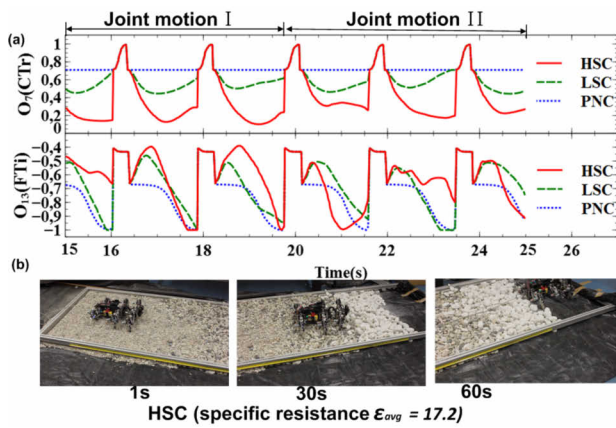


Figure 9: Comparisons of HSC, LSC and PNC for walking on gravel surface. (a) Control signals  $O_7$  and  $O_{13}$  for the CTr and FTi joints of the right front leg. There are six stance and five swing phases between 15 s and 25 s. (b) A series of photos shows hexapod robot walking controlled by HSC.

In addition to gravel surface, we also tested HSC, LSC and PNC for walking on another loose surface (i.e., snow), which has a thickness of 8cm. The experimental result also shows that HSC allows the hexapod robot to achieve more energy-efficient locomotion (see average specific resistance Table 1), compared to LSC. Note that we did not calculate average specific resistance of the hexapod robot controlled by PNC, since it got stuck in the snow. The experimental video of walking on snow surface can be seen at <https://www.youtube.com/watch?v=OkZiVNeQdCA>.

## Conclusion and Future Work

We implemented a neuromechanical controller on a hexapod robot for walking on sponge, gravel and snow surfaces. The controller coordinates 18 joints, generates basic locomotion, and allows for simply changing compliance of its legs for walking on the different surfaces. Due to the changeable

compliance, the robot can achieve more energy-efficient locomotion (i.e., lower specific resistance) on different surfaces. Softer legs (i.e., LSC setup) do better in locomotion on a soft elastic surface (i.e., sponge), while stiffer legs (i.e., HSC setup) are better for locomotion on loose surfaces (i.e., gravels and snow). In addition, on gravel surface, the specific resistance of the robot is 17.2 when it is controlled by the neuromechanical controller with HSC presented here. In contrast, its specific resistance increases to 56.63 when it is controlled by an adaptive neural locomotion controller presented in our previous work (Manoonpong et al., 2013), which does not have muscle-like mechanisms (i.e., virtual agonist-antagonist mechanism (VAAM)).

Central properties of the VAAM of our neuromechanical controller are: (1) it enables robot legs to simply change their compliance without the requirement of additional physically compliant mechanisms (Ham et al., 2009) or joint torque sensing (Görner and Hirzinger, 2010) and (2) it allows a hexapod robot to adapt its legs to deal with challenging surfaces (i.e., sponge, gravel and snow). In future work, we plan to compare the proposed neuromechanical controller with other adaptive leg controllers (e.g., forward model (Manoonpong et al., 2013)) in different surfaces (e.g., snow). And we will also implement an adaptive mechanism for automatically adjusting stiffness coefficients of the VAAM with respect to different walking speeds or gaits.

## Acknowledgement

This research was supported by the Emmy Noether Program of the Deutsche Forschungsgemeinschaft (DFG, MA4464/3-1), the Federal Ministry of Education and Research (BMBF) by a grant to the Bernstein Center for Computational Neuroscience II Göttingen (01GQ1005A, project D1), and European Communitys Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience. We also would like to thank Frank Hesse, Sakyasingha Dasgupta and Tomas Kulvicius for their fruitful discussions.

## Appendix A : Joint Motion Equation

Substituting Eqs.(7), (8) and (9) into Eq.(10), the motion equation of the joint  $P$  is given by:

$$I\ddot{\theta} = f^{ext} \sin(\theta)L + r[(K(l_1^P - l_0) + Dv_1^P + N_j i_1) - (K(l_2^P - l_0) + Dv_2^P + N_j i_2)]. \quad (23)$$

The lengths of  $PE_{(1,2)}$  (i.e.,  $l_{(1,2)}^P$ ) are equal to the lengths of  $M1$  (i.e.,  $L_1$ ) and  $M2$  (i.e.,  $L_2$ ),

$$l_1^P = L_1, L_2 = l_2^P. \quad (24)$$

In Fig. 4,  $M1$  is shortening when  $M2$  is lengthening. Therefore, the relationship between displacements of  $M1$  (i.e.,

$\Delta L_1$ ),  $M2$  (i.e.,  $\Delta L_2$ ) and  $PE_{(1,2)}$  (i.e.,  $\Delta l_{(1,2)}^P$ ) is given by:

$$-\Delta l_1^P = -\Delta L_1 = \Delta L_2 = \Delta l_2^P. \quad (25)$$

Here we postulate the relationship between displacements  $\Delta l_1^P$  of  $PE_1$ ,  $\Delta l_2^P$  of  $PE_2$  and the joint angle  $\theta$  as (derived by Eqs.(24) and (25)):

$$-(l_1^P - l_0) = -\Delta l_1^P = \theta r = \Delta l_2^P = l_2^P - l_0, \quad (26)$$

where  $r$  is the radius of the joint  $P$ . The relationship between velocities  $\dot{\Delta l}_1^P$  of  $PE_1$ ,  $\dot{\Delta l}_2^P$  of  $PE_2$  and the joint velocity  $\dot{\theta}$  is given by:

$$-v_1^P = -\dot{\Delta l}_1^P = \dot{\theta} r = \dot{\Delta l}_2^P = v_2^P. \quad (27)$$

Besides, since the motions of  $M1$  and  $M2$  are against each other, their activation intensities  $i_{(1,2)}$  are set to:

$$i_1 = -i_2 = 0.5. \quad (28)$$

## Appendix B : Notations

- $F_{6 \times 1}^{ext}$  is the matrix of the forces, i.e.,  $F_{6 \times 1}^{ext} = [f_1^{ext}, f_2^{ext}, \dots, f_6^{ext}]^T$ ;
- $F_{6 \times 1}^\perp$  is the Hadamard product of  $F_{6 \times 1}^{ext}$  and  $\sin(\theta 1_{6 \times 1})$ ;
- $L_1$  is the length of the link between the FTi joint and the end effector of the leg, e.g.,  $L_1 = 0.115$  m;
- $I$  is the inertia of the FTi and CTr joints, i.e.,  $I = 0.5 \times 10^{-3}$ ;
- $\ddot{\theta} 1_{6 \times 1}$  and  $\dot{\theta} 1_{6 \times 1}$  are the acceleration and velocity matrices of  $\theta 1_{6 \times 1}$ .  $r$  is set to 0.1;
- $K(1,2)_{6 \times 1}$  and  $D(1,2)_{6 \times 1}$  are matrices of the stiffness and damper coefficients of  $PE_{(1,2)}$ , which control the compliance of the FTi and CTr joints.
- $L_2$  is the length of links between the CTr and FTi joints, i.e.,  $L_2 = 0.075$  m;
- $\vec{V} 1_{6 \times 1}$  and  $\vec{V} 2_{6 \times 1}$  are matrices of the displacement vectors of the CTr and FTi joints relating to the forces  $f_{1-6}^{ext}$ .
- $\ddot{\theta} 2_{6 \times 1}$  and  $\dot{\theta} 2_{6 \times 1}$  are the acceleration and velocity matrices of  $\theta 2_{6 \times 1}$ .
- $C_{6 \times 1}$  is the matrix of the CTr neuron outputs of the MNN, i.e.,  $C_{6 \times 1} = [N_7, N_8, \dots, N_{12}]^T$ .

## References

Bermudez, F., Julian, R., Haldane, D., Abbeel, P., and Fearing, R. (2012). Performance analysis and terrain classification for a legged robot over rough terrain. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 513–519.

- Boaventura, T., Semini, C., Buchli, J., Frigerio, M., Focchi, M., and Caldwell, D. (2012). Dynamic torque control of a hydraulic quadruped robot. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1889–1894.
- Der, R. and Martius, G. (2012). The Ipzrobots simulator. In *The Playful Machine*, volume 15 of *Cognitive Systems Monographs*, pages 293–308. Springer Berlin Heidelberg.
- Galloway, K., Clark, J., Yim, M., and Koditschek, D. (2011). Experimental investigations into the role of passive variable compliant legs for dynamic robotic locomotion. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1243–1249.
- Görner, M. and Hirzinger, G. (2010). Analysis and evaluation of the stability of a biologically inspired, leg loss tolerant gait for six- and eight-legged walking robots. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4728–4735.
- Gregorio, P., Ahmadi, M., and Buehler, M. (1997). Design, control, and energetics of an electrically actuated legged robot. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 27(4):626–634.
- Ham, R., Sugar, T., Vanderborght, B., Hollander, K., and Lefeber, D. (2009). Compliant actuator designs. *Robotics Automation Magazine, IEEE*, 16(3):81–94.
- Lee, D. V., McGuigan, M. P., Yoo, E. H., and Biewener, A. A. (2008). Compliance, actuation, and work characteristics of the goat foreleg and hindleg during level, uphill, and downhill running. *Journal of Applied Physiology*, 104(1):130–141.
- Manoonpong, P., Parlit, U., and Wörgötter, F. (2013). Neural control and adaptive neural forward models for insect-like, energy-efficient, and adaptable locomotion of walking machines. *Frontiers in Neural Circuits*, 7(12).
- McCrea, D. A. and Rybak, I. A. (2008). Organization of mammalian locomotor rhythm and pattern generation. *Brain Research Reviews*, 57(1):134–146. Networks in Motion.
- Ozcan, O., Wang, H., Taylor, J., and Sitti, M. (2010). Surface tension driven water strider robot using circular footpads. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3799–3804.
- Qian, F., Zhang, T., chen Li, Hoover, A., Masarati, P., Birkmeyer, P., Pullin, A., Fearing, R., and Goldman, D. (2012). Walking and running on yielding and fluidizing ground. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia.
- Raibert, M. (2008). Bigdog, the rough-terrain quadruped robot. In Chung, M. J., editor, *Proceedings of the 17th IFAC World Congress, 2008*, volume 17.
- Saranli, U., Buehler, M., and Koditschek, D. E. (2001). Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631.
- Spence, A. J. (2011). Control strategies for legged locomotion: a comparative approach. In *7th European Nonlinear Dynamics Conference (ENOC), Rome, Italy*.
- Spence, A. J., Revzen, S., Seipel, J., Mullens, C., and Full, R. J. (2010). Insects running on elastic surfaces. *The Journal of Experimental Biology*, 213(11):1907–1920.

# ***FIMO*: Framework for Intrinsic Motivation**

Fabien Hervouet and Eric Bourreau

LIRMM

Université Montpellier 2

fabien.hervouet@lirmm.fr, eric.bourreau@lirmm.fr

## **Abstract**

This paper aims to introduce a flexible framework called *FIMO* dedicated to intrinsic motivation in developmental robotics. With this framework we want to offer a common way of implementing, testing and analyzing mechanisms related to intrinsically motivated algorithms. It may be seen as a generic complex open source framework for online exploration and structuring of learning spaces. We hereby lay both theoretical and practical foundations of our framework to encourage future experimental studies.

## **Introduction**

A commonly shared objective, in the developmental robotics community, as its name indicates, is insisting on the developmental part. It means that the most important aspect of any research led in this field lies in the path that leads to staged growth of learning. It may be about practical competences, conceptual comprehension or whatever a living organism with physical capabilities can practice. This implies that our goal is all about defining a model of developmental mechanisms, allowing a full open-ended incremental learning in an autonomous and interactive way, compliant and resilient with a real environment. The challenge is to propose a compromise between conciseness of the model and complexity it can generate as soon it is instantiated. But for an agent, developing competences is not that easy. It has to start with some coarse grained capabilities which are going to be refined throughout the developmental phase and life in general. This upgrowth is characterized by a double brain/body maturity, but also by the acquisition of sensorimotor experiences supporting learning. Inside the developmental robotics community, this is currently realized by inspecting multiple aspects of development. First the individual autonomous mental development which insists on finding mechanisms for an interactive system to develop itself solely using its very low-level sensory inputs and acting through its low-level motor outputs. The main second path for exploring development is the social learning. In this paper, we are interested in the first view.

Precisely, our working frame is mechanisms that can play the role of heuristics in order to focus and control the exploration of the potentially huge sensorimotor space of a situated and fully embodied agent. We believe that the goal of any developmental robotics algorithm is to design a control loop that – if executed on robot with physical capabilities – should reveal its own morphology affordance within its environment, through an unsupervised process. Moreover we also believe, as we already said, that the key challenge is to identify and implement low-level mechanisms that allow a long-term development as a scaffolding of capabilities. The more low-level these mechanisms are, the more the system can be considered as relevant. In the case of bio-inspired developmental robotics, we are interested in sensorimotor learning through the open-ended exploration of high-dimensional and complex bodies. This means we have to inspect and design scalable task-independent mechanisms that may involve the robot in a self autonomous skill practice. The main idea remains that the huge sensorimotor space can be divided into subspaces in order to facilitate the learning of the full space using intrinsic heuristics of space exploration. Another way to explain our view is to say that we are aiming at transferring the traditional cognitive modeling biases towards the natural limits and constraints imposed by the sensorimotor grounding embodiment.

In our case, we here underline the need for a framework for intrinsic motivation as we introduce it, with the capability to make parametric studies and introduce new bio-inspired ideas from state of the art neuroscience or developmental psychology research. We insist on the fact we propose in our framework *FIMO* facilitations for future improvements for some parts of the general intrinsically motivated algorithm it implements. In the rest of the article, we first propose a motivational background of the specific research field of ours. Then we present the practical foundations of this framework called *FIMO*. Then we zoom on the theoretical model of our view of intrinsic motivations. We conclude this paper by recalling the need for such a framework in our community, and draw some interesting application perspectives.

## Motivational Background

The question to ask when we get to implement some mechanisms like motivation in artificial agents, is about the origin of the notion of motivation. In this case, psychology has studied motivation a lot. The very first background of works about intrinsic motivation is the self-determination theory, which is the result of psychological research led by Deci and Ryan (1985) from the human point of view. This theory says that any individual has innate tendencies towards personal growth and vitality that are either satisfied in relation to their immediate environment. The theory also explains that there are three satisfactory needs that actors seek to satisfy: competence, relatedness and autonomy. Innate tendencies are enacted when these needs are fully satisfied. We will see that these founding psychological works have been at the base for transferring the notion of motivation *in silico*.

Historically, Schmidhuber (1991a) was the first to introduce research results about the importance of what he called *artificial curiosity*, in the sense of applying typical *human curiosity* to *artificial machines*. He argues that his research work has been driven by the simple idea that an optimal adapted and motivated agent is nothing but an agent trying to improve its compressed model of the world. He said that an agent with a prediction module, allowing the lossless compression of data by finding regularities in the world, is driven toward an optimal improvement. Schmidhuber (1991b) proposed to measure the learning progress of an agent by comparing difference of prediction for a same situation before and after the reality feedback. He therefore introduced the idea that an optimal curious agent's interest lies in the narrow corridor between what is simply too compressible and therefore uninteresting and boring, and what is not compressible at all because of a lack of regularity making it too complicated to learn.

Barto et al. (2004) have extended their own work in reinforcement learning (Sutton and Barto, 1998) with the notion of intrinsic motivation. They introduce an elaboration of the existing reinforcement learning framework that “encompasses the autonomous development of skill hierarchies through intrinsically motivated reinforcement learning”. Their model advocates the creation and use by an agent of structures of intrinsic generic rewards allowing adapted behavioral learning. They do not consider in any way external rewards.

Soon after the birth of the developmental robotics community (see Weng et al., 2001), Oudeyer and Kaplan (2004) introduced the *IAC* (*Intelligent Adaptive Curiosity*) algorithm. It explicitly refers to Schmidhuber's adaptive curiosity and must be linked with another paper published the same year by Steels (2004). The proposed mechanism is anchored at a sensorimotor level and allows low-level action selection in the high-dimensional sensorimotor space for a robot. This algorithm postulates that one way to provide

autonomy to a robot is to let it make its proper action choices, based on its experience, in order to maximize its learning. With this architecture an embodied agent is going to experiment grounded sensorimotor coordinations in order to learn the effects of its actions thanks to a unique action selection mechanism that tends to choose actions that improve prediction quality.

Baranes and Oudeyer (2010) then proposed an evolution to the original *IAC* algorithm called *SAGG-RIAC* which is of interest to us because of the competence acquisition paradigm it explores. The global principle remains the same except that this time, the agent has to choose sensory regions where it wants to return to, instead of choosing sensorimotor regions where it comes from. Practically, the *SAGG-RIAC* algorithm is based on alternating reaching phases (i.e. reaching a goal in what they call an operational space) and local exploration phases (i.e. improving the world comprehension toward the goal). The purpose of reaching phases is to test the reliability of the forward motor model while the purpose of exploration phases is to improve the inverse model of the system in the close vicinity of the current state. Exploration phases are triggered when the reliability of the local controller is too low. In the following section we explain some improvements to this algorithm we would like to introduce and experiment.

This research must also be linked with some other works by Blank et al. (2005) we fully agree with, where they argue that any intrinsic developmental algorithm has to be based on a recursive model that produces complex behavior and that it should rely on three concepts: abstraction, anticipation and intrinsic motivation.

## General Architecture of *FIMO*

*FIMO*<sup>1</sup> has the goal of bringing up a brand new open, flexible and extensible framework for research in the developmental robotics field working on agents driven by intrinsic motivations. We propose with this architecture, a solid and well-thought way of experimenting new ideas for the intrinsic motivation interested community. This section is dedicated to the presentation of *FIMO*, either the way it is architected, its pythonic foundations, its software architecture and workflow, and the default provided environments.

## Python Foundations

First of all, it is important to explain the Python foundations of this framework. We based our development on well documented, community supported and very powerful libraries such as *SciPy*, *NumPy*, *matplotlib* which are dedicated to scientific data processing. The first one Jones et al. (2001) is open-source software for mathematics, science, and engineering. It depends on the second one, which provides con-

<sup>1</sup>The Python open source code of the framework is available at <https://info-depot.lirmm.fr/republic/fimo> (public cloning repository) released under GNU GPLv3 license.



venient and fast  $N$ -dimensional array manipulation, with efficient numerical routines. The visualization module relies for its part on the third one Hunter (2007). Our point is that the techniques used in this framework have been already deeply tested and used by a lot of other researchers, and that they have proved their efficiency and their reliability.

### Software architecture

The global workflow of our framework *FIMO* is rather common and intuitive. Typically, the framework features an environment that describes in a formal way the being explored sensorimotor space, which is mandatory for bootstrapping the agent's living loop. Once the environment is chosen, we must also provide the agent with parameters and metrics – which may have a significant influence on its developmental trajectory – depending on the choices made at the instantiation time. Then, at the end of a run, i.e. when the program has tried to reach the certain amount of goals you asked it to perform, multiple data are dumped into log files. Typically, it includes the evolution of space partitioning into subregions, acquired sensorimotor raw data, interests of selected regions where objectives were generated and test results. Finally, another powerful advantage of the framework also lies in the visualization module made possible thanks to the logging process. This possibility to observe the result of any run instance makes it easier to understand and validate the configured parameters, metrics and other implemented ideas. We will illustrate the rest of the paper with some visualizations *FIMO* can offer.

### Environments

An environment as we defined and formalized it within our framework is all about the agent's acting possibilities. We mean that an environment should be seen as a complex interface between the sensory (or operational) space to explore, the motor space allowed for this exploration, and the combined result of these two. An environment must provide the sensory dimensions or operational dimensions, the motor dimensions, a starting state or rest position. It must also specify a function computing a new state given a current state and an action to execute, and a function that generates reachable coordinates in the sensory / operational space to be used for the examination. As a bias, you can predefine a region partitioning for the agent to start with, in order for instance to help it bootstrapping. The way we implemented the connection between environments and the main loop must allow to easily connect to some existing 3D simulated environments or to any physical robots following the same guidelines. Table 1 provides a summary of proposed simulated environments in *FIMO* and present their sensory and motor dimensions.

**Wheeled vehicle** The first provided environment is a very simple wheeled vehicle in an arena. The first version is only two dimensional (1D sensory space and 1D motor space).

| Environments                                     | a | b  |
|--------------------------------------------------|---|----|
| One-wheeled vehicle (OWV) 1D fully reachable     | 1 | 1  |
| OWV 1D not fully reachable                       | 1 | 1  |
| OWV 1D bump requiring sufficient inertia to pass | 1 | 1  |
| Two-wheeled vehicle (TWV) 2D squared area        | 2 | 2  |
| TWV 2D squared area with obstacles               | 2 | 2  |
| TWV 2D triangled area                            | 2 | 2  |
| Robotic arm (RA) with one joint                  | 2 | 1  |
| RA with two joints                               | 2 | 2  |
| RA with three joints                             | 2 | 3  |
| RA with fifteen joints                           | 2 | 15 |

Table 1: Summary of the variations of the main existing environments in *FIMO* presenting their distinctive feature(s), the size of their operational (a) and motor (b) spaces.

The second version is four dimensional (two sensory space, 2D motor space). Generally the shape of the arena is a square, but it could be different or contains obstacles. In the first version, the vehicle evolves in a 1D space and has a sole coordinate ( $x$ ), that is the distance to the front wall. In the second version, the vehicle evolves in a 2D space and has two coordinates ( $x; y$ ). It can move by performing an action ( $\Delta_x, \Delta_y$ ) representing a shift in the plane.

**Robotic Arm** We propose a second environment typically used – in particular in Baranes and Oudeyer (2010, 2013) – to test intrinsic motivation algorithms: a robotic arm (cf. figure 1<sup>2</sup>). The environment provided is a generic robotic arm that you can easily instantiate by defining the number of joints/limbs you want, and their respective length (or the unique length if ever).

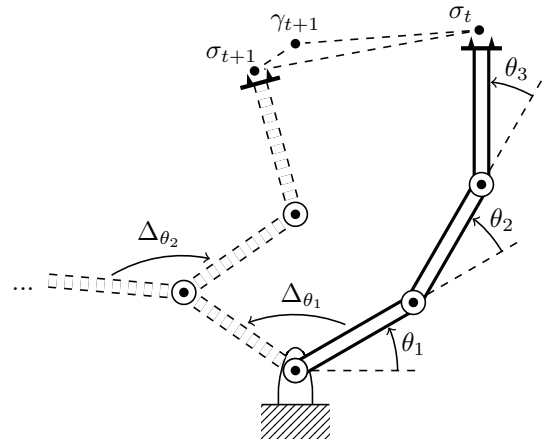


Figure 1: Schematic successive positions of a 3 joints robotic arm environment at time  $t$  and  $t + 1$ .

<sup>2</sup>Here, each limb has the same length and  $\theta_i$  represent angles of each joint, relatively to  $\theta_{i-1}$ . The performed action that moves the end-effector from  $\sigma_t$  to  $\sigma_{t+1}$  coordinates in the operational space by trying to reach  $\gamma_{t+1}$  was  $\alpha = (\Delta\theta_1 = +115, \Delta\theta_2 = -140, \Delta\theta_3 = +40)$ .

The transition from end effector position to another one through a given action is computed using standard methods for forward position kinematics. An action is a vector of positive or negative  $\Delta$  for each joint  $(\Delta_{\theta_1}, \Delta_{\theta_2}, \Delta_{\theta_3})$ . It means the agent can only use relative moves from one  $(H_{x,t}, H_{y,t})$  to the next  $(H_{x,t+1}, H_{y,t+1})$  position.

### Theoretical Model

In this section we explain the theoretical model behind of *FIMO*. Thus, we present the main algorithm and every single parameters or metrics that the experimenter must use as such or improve, extend or change. Nevertheless, what we want to emphasize is that this algorithm needs to be configured by many parameters and metrics to be specified by the experimenter that can strongly influence the final behavior of the system and its evaluation.

The living loop algorithm as we call it, proposes a kind of reinforcement learning method, but it must be seen as an empty shell that must be completed with the right parameters to be able to observe the best behavior (see algorithm 1). Indeed, in the same way that one must specify an environment, the configuration parameters and the choice of metrics are essential and grouped in a *config.py* file. The interaction of the body/environment and the intrinsically motivated algorithm is something very delicate which requires the attention of the experimenter. One must see this guided exploration of sensorimotor space as a *complex system* as it is composed with multiple interconnected parts (different parameters and metrics to tune) which as a whole exhibit behavior not obviously predictable from the individual properties. The interaction of these parameters put together in the living loop algorithm may exhibit emergent properties. Moreover we believe that it may be possible to define some precise optimized setting for a specific environment.

In the rest of the section we present some of the general part of the framework (main algorithm, default regions structure and implemented learning method) as well as the criticals part that may be overridden with improvements (competence and interest measures, the way memory may be restructured). These parameters should be deeply studied in order to fit the need for a specific morphology, because we believe that the embodiment plays a major role in order to help determining fully appropriate, responsive, efficient and developmental settings.

Thus this is all about trying to find a compromise to encourage and facilitate effective and efficient exploration of space, without losing time in not interesting areas, while allowing rapid improvement for test result. Among the default implementation choices we present here, some have already been presented in Hervouet and Bourreau (2012).

### Living loop algorithm

Although we keep the overall operation of the original motivational living algorithm SAGG-RIAC Baranes and Oudeyer (2010), we extended the frame. The global

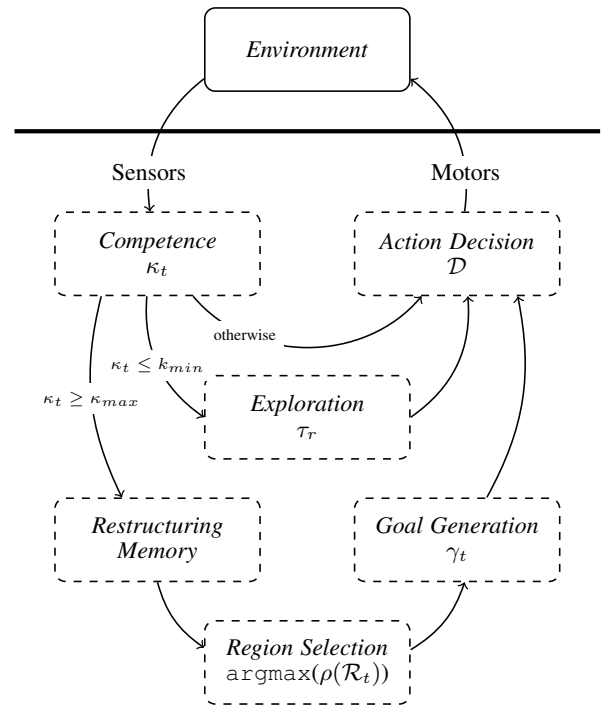


Figure 2: General algorithmic architecture of the intrinsically motivated living loop implemented in *FIMO*.

algorithmic architecture is presented in figure 2 and its implementation is presented in algorithm 1. With this implementation, we want to underline the flexibility of certain specific parts of the algorithm. We mean that the sequence of the algorithm is nothing but a generic system that aims at *exploring and structuring a sensory / operational space with partial data for sensorimotor learning*. Indeed we have to be aware of the strong influence of metrics and parameters that we may use to obtain more compliant or efficient behavior. This is the reason why we decided to emphasize the critical parts in the algorithm in order to be more illustrative. Each of them is studied in the rest of this section.

In a formal way, our architecture makes a distinction between *raw data*  $\xi_r$  accumulated during *exploration phases* (second loop in fig. 2) and more *structured data*  $\xi_g$  accumulated during *exploitation phases* (third loop in fig. 2). An exploration phase is triggered each time the agent is considered as incompetent and consist in accumulating  $\tau_r$  raw data about the consequences of actions the agent executes. In practice, exploration data represent a forward model of the body in the world. Exploitation phases consist in generating a goal and attempting to reach it in  $\tau_g$  times. Exploitation data represent the particular historical and motivational coupling between the agent and its environment, in the sense of Varela et al. (1991), i.e. the sensory configuration he has set itself for its goals that it tried to achieve by itself with varying degrees of success, according to the increase of its competence in relation to these goals.

As it is implicitly formalized, time is considered in the algorithm:  $\sigma_t$  always represents the agent's current state, which changes every time the agent performs an action  $\alpha_t$  using the *execute*( $\alpha$ ) method.

---

**Algorithm 1:** Intrinsically Motivated Living Loop
 

---

**input:**  $\sigma$  experienced states;  
 $\mathcal{R}$  set of existing regions;  
 $\gamma$  set of self generated goal;  
 $\rho$  interest measure;  
 $\kappa$  competence measure;  
 $\mathcal{D}$  action decision method;  
 $\xi_r, \xi_g$  respectively raw and goal experiments;  
 $\tau_r, \tau_g$  resp. exploration and reaching trials;

```

1 while True do
2    $\sigma_{start} \leftarrow \sigma_t$ 
3    $\mathcal{R}_t \leftarrow \operatorname{argmax}(\rho(\mathcal{R}_i) \forall \mathcal{R}_i \in \mathcal{R})$  (see section IM)
4    $\gamma_t \leftarrow \operatorname{randomGoal}(\mathcal{R}_t)$ 
5    $\mathcal{A} \leftarrow \emptyset$ 
6   repeat
7      $\alpha_i \leftarrow \mathcal{D}(\xi_r, \sigma_t, \gamma_t)$  (see section AD)
8      $\mathcal{A} \leftarrow \mathcal{A} \cup \alpha_i$ 
9     execute( $\alpha_i$ )
10     $\kappa_t \leftarrow \kappa(\sigma_{start}, \gamma_t, \sigma_t)$  (see section CM)
11     $\xi_r \leftarrow \xi_r \cup (\sigma_{t-1}, \alpha_i, \sigma_t)$ 
12    if  $\kappa_t \leq \kappa_{min}$  then
13       $\xi_{g, \mathcal{R}_{\sigma_t}} \leftarrow \xi_{g, \mathcal{R}_{\sigma_t}} \cup (\sigma_{t-1}, \sigma_t, \{\alpha_i\}, \sigma_t, 0)$ 
14      repeat
15         $\sigma_t \leftarrow \sigma_t - 1$ 
16         $\alpha_j \leftarrow \operatorname{randomAction}()$ 
17        execute( $\alpha_j$ )
18         $\xi_r \leftarrow \xi_r \cup (\sigma_{t-1}, \alpha_j, \sigma_t)$ 
19      until  $\tau_r$  trials not exceeded
20    end
21  until  $\kappa_t \leq \kappa_{max}$  or  $|\mathcal{A}| \leq \tau_g$ 
22   $\xi_{g, \mathcal{R}_t} \leftarrow \xi_{g, \mathcal{R}_t} \cup (\sigma_{start}, \gamma_t, \mathcal{A}, \sigma_t, \kappa_t)$ 
23  restructuringMemory() (see section RM)
24 end

```

---

### Interest Measure (IM)

The measure of interest  $\rho$  qualifies the dynamic interest of a region. It is used to compute the most interesting region, the one with maximum  $\rho$  value (line 3), the agent is going to self-generate a goal in. The default one proposed in *FIMO* is based on the one introduced by Baranes and Oudeyer in Baranes and Oudeyer (2010) with one major difference. It is computed as follows:

$$\rho(\mathcal{R}_i) = \text{learningProg}(\mathcal{R}_i) + \text{diversification}(\mathcal{R}_i)$$

The learning progress is computed using experiments from exploitation phases ( $\xi_g$  for goal experiments) which are of the form:

$$\xi_{g,t} = (\sigma_t, \gamma_t, \mathcal{A}, \sigma_{t+i}, \kappa_t)$$

with  $\sigma_t$  the current sensory configuration state,  $\gamma_t$  the chosen goal to be reached,  $\mathcal{A}$  the motor configurations successively performed to achieve the goal with  $|\mathcal{A}| = i$ ,  $\sigma_{t+i}$  the new current state after execution of actions and the competence  $\kappa_t$  (explained in detail in a following section). The learning progress computation can be seen as a derivative of competences. Let  $\kappa_j$  be the competence of the  $j^{th}$  experiment stored in memory, and  $|\mathcal{R}_i|$  the number of experiments in a region.

$$\text{learningProg}(\mathcal{R}_i) = \frac{|\sum_{j=0}^{|\mathcal{R}_i|/2} \kappa_j - \sum_{j=|\mathcal{R}_i|/2}^{|\mathcal{R}_i|} \kappa_j|}{|\mathcal{R}_i|}$$

We chose to directly incorporate a UCT based Kocsis and Szepesvári (2006) diversification measure which takes into account in an incremental way the number of experiments conducted in the current region relative to the total number of experiments. It means that this mechanism will gently wake up regions whose direct learning progress is decreasing but for which it would be wise to generate a goal in just to make sure everything has been already either understood or misunderstood. Let  $n$  and  $n_i$  be respectively the total number of experiments and the number of experiments in the current region. Let  $c$  be a constant that allows to normalize the result of diversification measure.

$$\text{diversification}(\mathcal{R}_i) = c \times \sqrt{\frac{\ln n}{n_i}}$$

In summary, the interest measure  $\rho$  is composed by the addition of two dynamic but separate measures. The first one computes the learning progress in a given region, while and the second one tries to uniformly balance against the natural excessive regional intensification of the first one. This separation of the intensification measure and the diversification measure is of real interest for future improvements, because we will only have to deal with precisely defining the first one without taking into account the uniformisation process preventing overspecialization.

### Action Decision (AD)

The default decision method we propose is a simple algorithm that chooses the next action to perform in order to reach a goal driven through experience (line 7). We propose to compute action towards a goal using *k-nearest-neighbour* experiments chosen among previously acquired explorative experiments. These experiments ( $\xi_r$  for raw experiments) are of the form:

$$\xi_{r,t} = (\sigma_t, \alpha_t, \sigma_{t+1})$$

with  $\sigma_t$  the current sensory configuration state,  $\alpha$  the motor configuration determined at time  $t$  and finally  $\sigma_{t+1}$  the new current state after execution of action  $\alpha$ . These experiments must maximize two criteria: the initial and final states should be as close as possible respectively to current and goal states. The strategy then consists in generating a mean

action with respect to actions performed in these filtered experiments (with  $|K| = k$ ).

$$\mathcal{D}(\xi_r, \sigma_t, \gamma_t) = \frac{\sum_{u=1}^{|K|} \alpha_u}{k} \text{ where}$$

$$K = \{\xi_{r,u} \in \xi_r : \min(|\sigma_t - \sigma_u| + |\gamma_t - \sigma_{u+1}|)\}$$

### Competence Measure (CM)

Competence  $\kappa_t$  ranges in  $[\kappa_{min}; 0]$ , as it is computed in the original algorithm, with  $\kappa_{max}$  typically equal to  $-1$ . The default implemented measure sums distances for all sensors in  $S$  between the final sensory position minus the goal sensory position, relative to the start position minus the goal position. It means that we compute the shift between where the agent comes from, where it should have arrived and where it has finally arrived.

$$\kappa(\sigma_s, \gamma_t, \sigma_f) = \max\left(-\frac{|\sigma_f - \gamma_t|}{|\sigma_s - \gamma_t|}, \kappa_{min}\right)$$

with  $\sigma_s$ ,  $\gamma_t$  and  $\sigma_f$  respectively start, goal and final state  $\kappa_{min}$  the minimal competence (or maximal incompetence) and  $\kappa_{max}$  the very competent threshold. The 3D visualization of the competence value of the robotic arm evolving in a 2D space is presented in figure 3. We can observe the reachable area at the center.

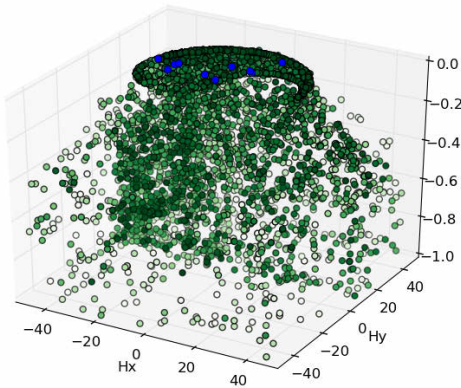


Figure 3: 3D visualization of generated goals for an instance run of the robotic arm in the two dimensional operational space. The closer to 0 on the third axis the goals, the more competent the agent was to reach them. The greener the goals, the more recently they have been reached. Blue points represent goals for assessment.

### Restructuring Memory (RM)

A typical developmental and incremental process pushes the robot to start its learning from scratch. It can only make non-optimal decisions solely from previously acquired data. This implies that the robot acts very strangely at

the beginning because it does not hold enough information about the world it develops in. The figure 4 presents the result of a run instance with original operational space split into subregions of interest through time<sup>3</sup>. We can observe the naturally delimited reachable space because we add perfectly reached goals from where we arrive at the end of a reaching attempt. Moreover we can observe the reinforcement of generated goals in the vicinity of these natural embodiment limits. This shows that the agent seems to find interesting areas located near the limits it can reach, which must be considered as a good behavior.

Therefore, pushing the robot to split its sensorimotor space allows it to overcome the lack of information at the beginning of the developmental living process. This is this particular splitting condition that makes the strength of this approach because it allows *the isolation of coherent experiments*. This coherence may depend on the nature of the measure that determines the best split in a sensorimotor region. The default coherence measure implemented in *FIMO* is related to the notion of learning progress<sup>4</sup>, i.e. the derivative of learning.

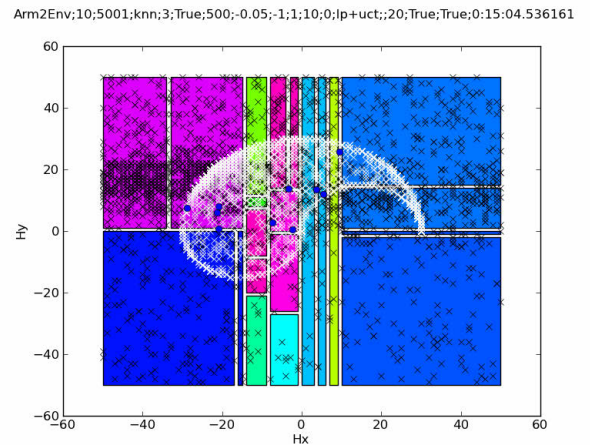


Figure 4: Visualization of generated experienced goals and regions containing them in the split operational space for a robotic arm environment.

Thus the splitting condition needs to take into account knowledge and experience accumulated. Otherwise the agent will always tend to split again and again its space, which will probably be pointless. We wanted to propose a general way to be able to implement a kind of dual restructuring measure. Because it may be very interesting to

<sup>3</sup>The shape of the reachable area is not a full circle because of the limitation we added for each joint's absolute angle  $\theta_i$  to range between  $[0; 180]$ .

<sup>4</sup>But it could be related to any other idea that we could think of progress: better structuring memory, novelty, compression, etc. This is the reason why we should facilitate the implementation of new ideas by emphasizing replaceable metrics and parameters.



be able to elaborate a measure that would decide, depending on accumulated experience, not only how to split, but more how to reorganize the memory. This vision totally fits a developmental process. Thus we chose to combine these two mechanisms by proposing by default a dual measure for both splitting and merging. It tries to maximize the absolute value of the difference between the learning progress in the two subregions relative to the current learning progress in the mother region.

$$\mu(\mathcal{R}_1, \mathcal{R}_2, R) = \frac{|LP(\mathcal{R}_1) - LP(\mathcal{R}_2)|}{LP(R)}$$

It means that in the splitting case we will split the current region  $R$  if it contains two subregions  $\mathcal{R}_1$  and  $\mathcal{R}_2$  exhibiting better learning progress, relatively to the current learning progress in the region  $R$ . On the other hand in the merging case, we will merge two regions that have been split possibly a long time ago when the agent didn't hold enough information: the current region  $\mathcal{R}_1$  should be merged with another region  $\mathcal{R}_2$  into a sole region  $R$  if the learning progress of  $R$  exhibits a better evolution than  $\mathcal{R}_1$  and  $\mathcal{R}_2$ .

### Regions as a graph structure

In order to propose more genericity for future implementations we chose to upgrade the tree structure of regions used since *IAC* Oudeyer and Kaplan (2004). We decided to introduce a graph structure of regions (cf. figure 5) because we considered that the tree representation was a limitation. We believe that, by its more scalable and flexible nature, the graph structure facilitates from an operational point of view potential reorganization as we will discuss in the following section. In particular it may allow the creation of non-convex regions by merging already existing region whether they are physically adjacent or not.

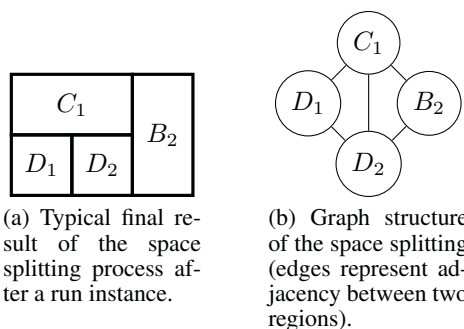


Figure 5: Illustration of the graph structure for regions. On the left it represents the result of a 2-dimensional sensory space splitting process, while on the right the way this is handled in memory.

### So, how to evaluate?

To propose a framework laying the foundations to encourage the community to easily implement and test new ideas is one thing, to propose a way to evaluate them is another. Indeed setting up such a framework obviously also involves setting

up an experimental validation process as a common foundation for evaluation, which is not a trivial thing. As pointed out by Meeden and Blank (2006) a few years ago, the evaluation question is fundamental, especially in the frame of autonomous developmental robotics.

We generally make a distinction between the formative and the summative assessments in the literature. A quotation attributed to Robert Stakes in Scriven (1991) tends to explain the difference between them: "When the cook tastes the soup, that's formative; when the guests taste the soup, that's summative".

*Formative assessment* represents the global operation of the intrinsically motivated process proposed living loop algorithm. Indeed, the main role of formative assessment lies in its subjective regulatory function of the learning process within the system, where the learner must be able to measure progress made and progress to be made.

*Summative assessment* must be seen as a more common and accepted way permitting to test learning acquisition from outside of the system, by certifying whether knowledge has been assimilated. This means we must provide in *FIMO* an external way, considered as objective as possible, to measure the relevance and importance of potential improvements as explained in Hervouet (2013). Practically in *FIMO* we implemented a measure that circumscribes the evaluation of the evolution of the competence through time. The idea is to select a set of arbitrary uniform reachable goals and compute a mean incompetence for reaching them.

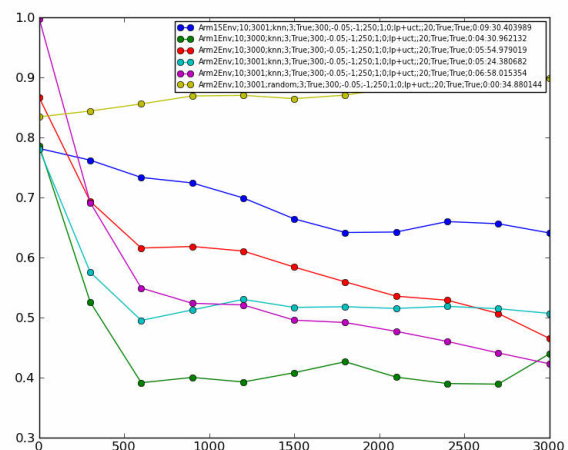


Figure 6: Evolution of the mean incompetence for a given set of reachable goals for different environment and parameters (1 means bad competence while 0 means very competent).

As an illustration, we propose on figure 6, a comparison of assessment scores for a set of different settings under different environments. From bottom: (1) 1-joint robotic arm; (2,3,4) three instantiations with the same settings of a 2-

joints robotic arm; (5) 15-joints robotic arm; (6) 2-joints robotic arm but with random action (baseline).

## Conclusions & Perspectives

In this paper we argue that in the developmental robotics community, and more specifically in the subcommunity interested in intrinsically motivated robotics, there are multiple ways to propose some improvements. With the proposed framework *FIMO*, we clearly made an attempt, starting from an existing approach, to extend the frame. It is absolutely not the purpose of this article to introduce some fascinating new metrics or define more precisely some of the parameters. We are not saying we have found the best something or a better somewhat. In contrast, we believe that this will be the purpose with forthcoming publications, thanks to the implementation of our framework and the facilitations it brings up for the evaluation and comparison of future improvements. Notwithstanding, there are various parameters waiting to be tuned. However, the fact remains that, to illustrate this purpose, we proposed some default parameters, metrics or even other surroundings learning features, but they are presented as an indication because we were confronted to the need to set up the framework with implementation choices.

We consider *FIMO* as a necessary formal step toward an open future of intrinsic motivations work because it will help future contributions and improvements to emerge. Our willingness for genericity, must be seen as a contribution in itself, in the sense of the facilitation for novel contributions.

Beyond this evident aspect, we could draw an interesting future for our framework. Although defining metrics is typically human compatible (and especially computer incompatible), simple parameters should not be set manually. As a spreading field, evolutionary robotics provides some exciting opportunities for the developmental robotics community, and could help us in this context. We mean that the very practical way of implementing evolutionary mechanisms consists in growing individuals with different genomes, and to make reproduce the best adapted ones in order to grow the next generation. This process can be reiterated as long as you may observe some interesting progress towards adaptation to the environment. That is why we truly believe that a possible *EvoFIMO* could constitute a very interesting perspective for the development of our research as well as for the whole community.

## References

- Baranes, A. and Oudeyer, P.-Y. (2010). Intrinsically-motivated goal exploration for active motor learning in robots: A case study. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*.
- Baranes, A. and Oudeyer, P.-Y. (2013). Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73.
- Barto, A. G., Singh, S., and Chentanez, N. (2004). Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of International Conference on Developmental Learning (ICDL)*. MIT Press, Cambridge, MA.
- Blank, D., Kumar, D., Meeden, L., and Marshall, J. (2005). Bringing up robot: Fundamental mechanisms for creating a self-motivated, self-organizing architecture. *Cybernetics and Systems*, 36.
- Deci, E. L. and Ryan, R. M. (1985). *Intrinsic Motivation and Self-Determination in Human Behavior (Perspectives in Social Psychology)*. Perspectives in social psychology. Plenum Press.
- Hervouet, F. (2013). Autour de la preuve en intelligence artificielle. In *La preuve et ses moyens – Actes des journées interdisciplinaires de Rochebrune (sous presse)*.
- Hervouet, F. and Bourreau, E. (2012). Improvement proposals to intrinsically motivational robotics. In *Proceedings of the second joint conference ICDL-Epirob'12*.
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95.
- Jones, E., Oliphant, T., Peterson, P., et al. (2001). Scipy: Open source scientific tools for python.
- Kocsis, L. and Szepesvári, C. (2006). Bandit based monte-carlo planning. In *ECML-06. Number 4212 in LNCS*, pages 282–293. Springer.
- Meeden, L. and Blank, D. (2006). Editorial: Introduction to developmental robotics. *Connection Science*, 18(2):93–96.
- Oudeyer, P.-Y. and Kaplan, F. (2004). Intelligent adaptive curiosity: a source of self-development. In Berthouze, L., Kozima, H., Prince, C. G., Sandini, G., Stojanov, G., Metta, G., and Balkenius, C., editors, *Proceedings of the 4th International Workshop on Epigenetic Robotics*, volume 117, pages 127–130. Lund University Cognitive Studies.
- Schmidhuber, J. (1991a). Curious model-building control systems. In *Proceedings of the International Joint Conference on Neural Networks, Singapore*, volume 2, pages 1458–1463. IEEE press.
- Schmidhuber, J. (1991b). A possibility for implementing curiosity and boredom in model-building neural controllers. In Meyer, J. A. and Wilson, S. W., editors, *Proceedings of the International Conference on Simulation of Adaptive Behavior*, pages 222–227. MIT Press/Bradford Books.
- Scriven, M. (1991). *Evaluation Thesaurus*. SAGE Publications.
- Steels, L. (2004). The autotelic principle. In *Embodied Artificial Intelligence*, pages 231–242. Springer.
- Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.
- Varela, F., Thompson, E., and Rosch, E. (1991). *The Embodied Mind: Cognitive science and human experience*. MIT Press, Cambridge.
- Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., and Thelen, E. (2001). Autonomous mental development by robots and animals. *Science*, 291.

# A practical approach to Humanoid Introspection

Ignazio Infantino<sup>1</sup>, Giovanni Pilato<sup>1</sup>, Riccardo Rizzo<sup>1</sup> and Filippo Vella<sup>1</sup>

<sup>1</sup> ICAR - CNR, Viale delle Scienze ed.11, Palermo, Italy  
ricrizzo@pa.icar.cnr.it

## Abstract

We describe an approach for a humanoid robot to understand its internal state (Infantino et al. (2013)). The method is based on self observation and communication with the external world, according to the idea of introspection given by Sloman (2010). The robot introspection arises from information about physical components and software modules. This information is translated in a spatial representation of the hardware and software components of the robot through a SOM, which links the state representation of the robot with an high level representation given by an ontology. The ontology is furthermore linked to a linguistic module that makes it possible the interaction with human beings through a conversational agent.

## Introduction

The interaction between man and robot can benefit from models of the human mind and its cognitive abilities. One aspect that has not received enough attention is the mechanism that allows the robot to have a kind of self-awareness (Birlo and Tapus, 2011). Self-awareness, including the body physical parts and management functions related to them, is often referred to an a priori knowledge and it is usually implicitly integrated into the architecture. We will illustrate a possible approach for the realization of an introspective capacity (at present mainly oriented to the physical components of the robot).

Based on an empirical approach, our idea of introspection of the robot, starts from the analysis of information obtained automatically by the embedded software and its related documentation, in particular regarding the relationship (direct or explicit) between the physical components and software modules. This documentation is used to construct a representation of the hardware and software of the robot on a map, based on Self Organizing Map (SOM), similar to the human somatosensory map. This map structure can be used to quickly retrieve information semantically related (Honkela et al., 1997). From this map we can use different approaches to ascend to a higher level of abstraction. In particular, the approach used in this paper involves a simple association be-

tween labels arising from data and ontology entities, trying to get the expressiveness with a enormous knowledge based on common sense using Cyc (see [www.opencyc.org](http://www.opencyc.org)).

## The Introspection Architecture System

Our approach to introspection is based on self observation and communication. Figure 1 shows the proposed architecture. Considering the definition of Sloman, self observation is what the robot should do in order to build, represent and understand its internal state. In particular it is necessary to have a set of sub-systems dedicated to make a snapshot of the Nao robot state. Some systems are supplied by Nao system software, while some others are developed ad-hoc. The data obtained are used to build a rich state representation that should be supplied to an ontology that associates a meaning to the internal state. Our approach integrates static and dynamic information on robot operation.

Static information is related mainly to robot hardware parts and software modules, like hardware drivers, or modules that supplies some services like face tracking. But these parts can be active or not during robot operation, and their state is part of the robot state itself. A simple list of these parts can be difficult to manage. In order to obtain a manageable state representation we decided to develop a map that collects all the robot parts, and to highlight on this map all the parts that are involved in any robot operation. Such a map can be obtained using the information contained in the robot documentation that is rich of hardware and software details. Dynamic information is also represented on the map but they came from robot operation. When the robot is operating it is possible to highlight on the map the units corresponding to the active modules and the unit corresponding to the hand. This is a part of the robot state representation. According to figure 1 this Nao State Representation is supplied to a Semantic Bridge that analyzes the representation and gives as output a set of information (semantic labels) that are used to activate the right concepts on the ontology.

The Linguistic Level exploits these activations in order to perform a verbal interaction with the human user. In the present implementation the Semantic Bridge is constituted

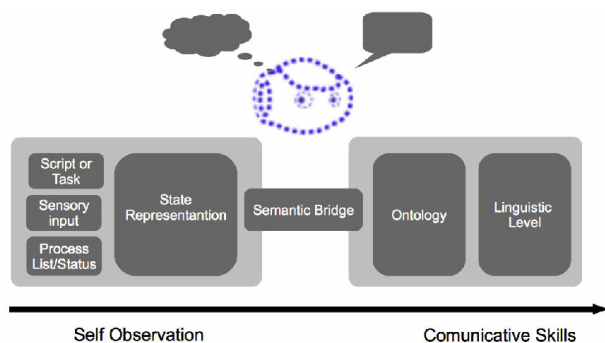


Figure 1: The proposed architecture for introspection

by a set of labels.

### Self Observation

The Nao state representation is obtained by using the information in documentation, the list of processes running on Nao system, and the information from sensor input. The Self Observation module builds a representation of the internal state of the robot, mixing together the information related to the robot hardware and the information available about software modules running during robot operation. This representation should be rich enough to represent the perception of the robot body from sensors, to distinguish different kind of sensors, and also to describe the list of active processes that control the robot movements and any other useful state component. The robot documentation collects information about the robot hardware and software, so it is a good starting point for this kind of representation. In order to obtain a suitable document representation, useful for clustering, a sequence of standard pre-processing techniques were applied. A SOM (Kohonen, 1995) architecture has been used to cluster semantically related documents. The sensory input receives the visual information from the cameras and filters the input to select the most promising areas for the downstream tasks. Nao software platform supplies a list of processes available on the Nao system. These scripts are also mapped in the document SOM using a suitable set of keywords. Finally a third part of the Nao state is obtained using some custom scripts that output the list of processes running on the Nao system.

### The Semantic Bridge and Linguistic Level

The semantic bridge subsystem obtains a set of suitable labels from the Nao internal state representation. At present the implementation of the semantic bridge uses a look-up table that connects the map state images to the labels in the ontology concepts.

The Nao robot has an internal knowledge of its physical structure and functionalities. This knowledge can be exploited to support direct communication on the perception capabilities of the robot, and to describe his state to an

human interlocutor by using natural language. Moreover, modern semantic tools and introspection capability can be exploited together to improve and support direct communication on the robot perception mechanisms (Infantino et al., 2012; Augello et al., 2013).

The Cyc knowledge base (KB) has been used to code relations, concepts, constraints, and rules regarding the Nao robot domain. These concepts have been organized in order to fulfil the self-observation task.

The linguistic level is aimed at interpreting natural languages query given by the user. This level exploits a classical pattern-matching technique enhanced with Cyc ontology inference capability. This feature is obtained by transforming natural languages requests into symbolic queries, expressed in the ontology language. Such commands are forwarded to the ontology engine that computes the appropriate inferences and gives results in a symbolic form. The symbolic answers are then transformed by the linguistic module into natural language sentences that are finally shown to the user. The linguistic level has been implemented by using the A.L.I.C.E. web bot (see [www.alicebot.org](http://www.alicebot.org)).

### References

- Augello, A., Infantino, I., Pilato, G., Rizzo, R., and Vella, F. (2013). Introducing a creative process on a cognitive architecture. *Biologically Inspired Cognitive Architectures*, available on line at: [www.sciencedirect.com/science/article/pii/S2212683X13000467](http://www.sciencedirect.com/science/article/pii/S2212683X13000467).
- Birlo, M. and Tapus, A. (2011). The crucial role of robot self-awareness in hri. In *Proc. of HRI 2011*, pages 115–116. ACM.
- Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. (1997). Websom - self-organizing maps of document collections. In *Neurocomputing*, pages 101–117.
- Infantino, I., Pilato, G., Rizzo, R., and Vella, F. (2012). I Feel Blue: Robots and Humans Sharing Color Representation for Emotional Cognitive Interaction. In *Biologically Inspired Cognitive Architectures 2012*, volume 196, pages 161–166.
- Infantino, I., Pilato, G., Rizzo, R., and Vella, F. (2013). A practical approach to humanoid introspection. *Intl. J. of Advanced Robotic Systems.*, 10:246.
- Kohonen, T. (1995). *Self-Organizing Maps*. Springer, Berlin.
- Sloman, A. (2010). An alternative to working on machine consciousness. *International Journal of Machine Consciousness*, 02(01):1–18.



# A Situated and Embodied Model of Ant Route Navigation

Andrew Philippides<sup>1</sup>, Bart Baddeley<sup>1</sup>, Philip Husbands<sup>1</sup>, Paul Graham<sup>1</sup>

<sup>1</sup> Centre for Computational Neuroscience and Robotics, University of Sussex, Brighton, UK

andrewop@sussex.ac.uk

## Abstract

This abstract summarises a model of route navigation inspired by the behaviour of ants presented fully in Baddeley et al. (2012). The ant's embodiment coupled with an innate scanning behaviour means that robust route navigation can be achieved by a parsimonious biologically plausible algorithm.

The ability of social insects to learn long foraging routes guided by visual information (Wehner, 2009) shows that robust spatial behaviour can be produced with limited neural resources (Chittka and Skorupski, 2011). As such, social insects have become an important model system for understanding the minimal cognitive requirements for navigation and, more generally those studying animal cognition using a bottom-up approach to the understanding of natural intelligence (Wehner, 2009, Shettleworth, 2010) while also providing inspiration for biomimetic engineers. Models of visual navigation that have been successful in replicating place homing are dominated by snapshot-type models where a single view of the world as memorized from the goal location is compared to the current view in order to drive a search for the goal (Cartwright and Collet, 1983; for review, see Möller and Vardy, 2006). Snapshot approaches only allow for navigation in the immediate vicinity of the goal however, and do not achieve robust route navigation over longer distances (Smith et al., 2007). Here we present an embodied parsimonious model of visually guided route learning that addresses these issues (Baddeley et al., 2012). By utilising the interaction of sensori-motor constraints and observed innate behaviours we show that it is possible to produce robust behaviour using a learnt holistic representation of a route. Furthermore, we show that the model captures the known properties of route navigation in desert ants.

Our navigation algorithm consists of two phases (see Baddeley et al., 2012, for details). The agent first traverses the route in 4cm steps with direction determined by a combination of noisy path integration (PI; true heading plus Gaussian noise, mean 0, s.d. 5°) and obstacle avoidance, during which the training views used to learn the route are experienced (a view is used after every 4cm step). In some experiments, a predefined learning walk is added to the start of the training path with training views taken every 2cm. To navigate, the agent visually scans the world by rotating on the spot through  $\pm 90^\circ$  of the current heading in  $1^\circ$  steps, behaviour similar to that observed in ants (P. Graham, *Personal Observation*). The most familiar direction during the scan is identified by inputting each view into an artificial neural network (ANN) trained to perform familiarity discrimination using the training views. Views are panoramic in azimuth and cover  $68^\circ$  of elevation above the horizon. Acuity is  $4^\circ$  meaning views are  $90 \times 17$ . The ANN is fully connected with  $90 \times 17$  (one per pixel) inputs and outputs and

no hidden layer. Weights are adjusted once per training view using an Infomax learning rule, with training views then discarded. The algorithm thus 'learns' routes after a single journey and memory load does not scale with route length. After training, the ANN outputs a familiarity score for each view input during a scan. Gaussian noise (mean 0, s.d.  $15^\circ$ ) is added to the direction associated with the most familiar view, a 10cm step is made in this direction, and the scanning routine repeats, until within 4cm of the goal or timed out.

We test our route navigation by learning a series of routes through visually cluttered environments consisting of objects distinguishable only as silhouettes against the sky. The model's performance is shown in figure 1. The model is able to learn idiosyncratic routes after a single training run (fig. 1A). As with ants, the routes show clear polarity and can only be traversed from start to goal. While successful for route navigation, if the agent misses the goal, it will typically continue in a direction similar to the last steps of the training route (fig. 1B) rather than search for it. To learn how to return to a specific goal location from nearby local surrounding regions, some ants perform a learning walk consisting of several loops out and back towards the goal when they first leave it. Adding such a walk to the training path means that when the agents nears or passes the goal, familiar directions are set by views experienced during the learning walk and draw the agent to the goal (fig. 1C). The model thus exhibits both place-search and route navigation with one mechanism. The model also learns multiple idiosyncratic routes to a goal (Fig. 1D-E). Here, three different routes are used but encoded within the same network which does not separate routes into distinct paths but stores all the information holistically. Given this performance, we believe our model represents the only detailed and complete model of insect route guidance to date.

Our approach is differentiated from previous attempts to understand route navigation in insects in several ways. 1) Navigation is independent of odometric or compass information. Unlike most snapshot-type models, training views are used as they are experienced, and are not rotated into common orientation before use. 2) The algorithm does not specify when or what to learn, but uses all views experienced during training. 3) Training views are not discrete waypoints. Previous route navigation algorithms navigate from one waypoint to another in a sequence, meaning one needs to know which waypoint is being used. Here, we do not navigate to each training view. Rather training views recall familiar directions not discrete places and are used to learn a holistic representation of the route; this representation says "What should I do?" not "Where am I?". This means that 4) navigation proceeds through a simple embodied strategy of rotating on the spot – a behaviour observed in navigating ants – and moving in the most familiar direction.

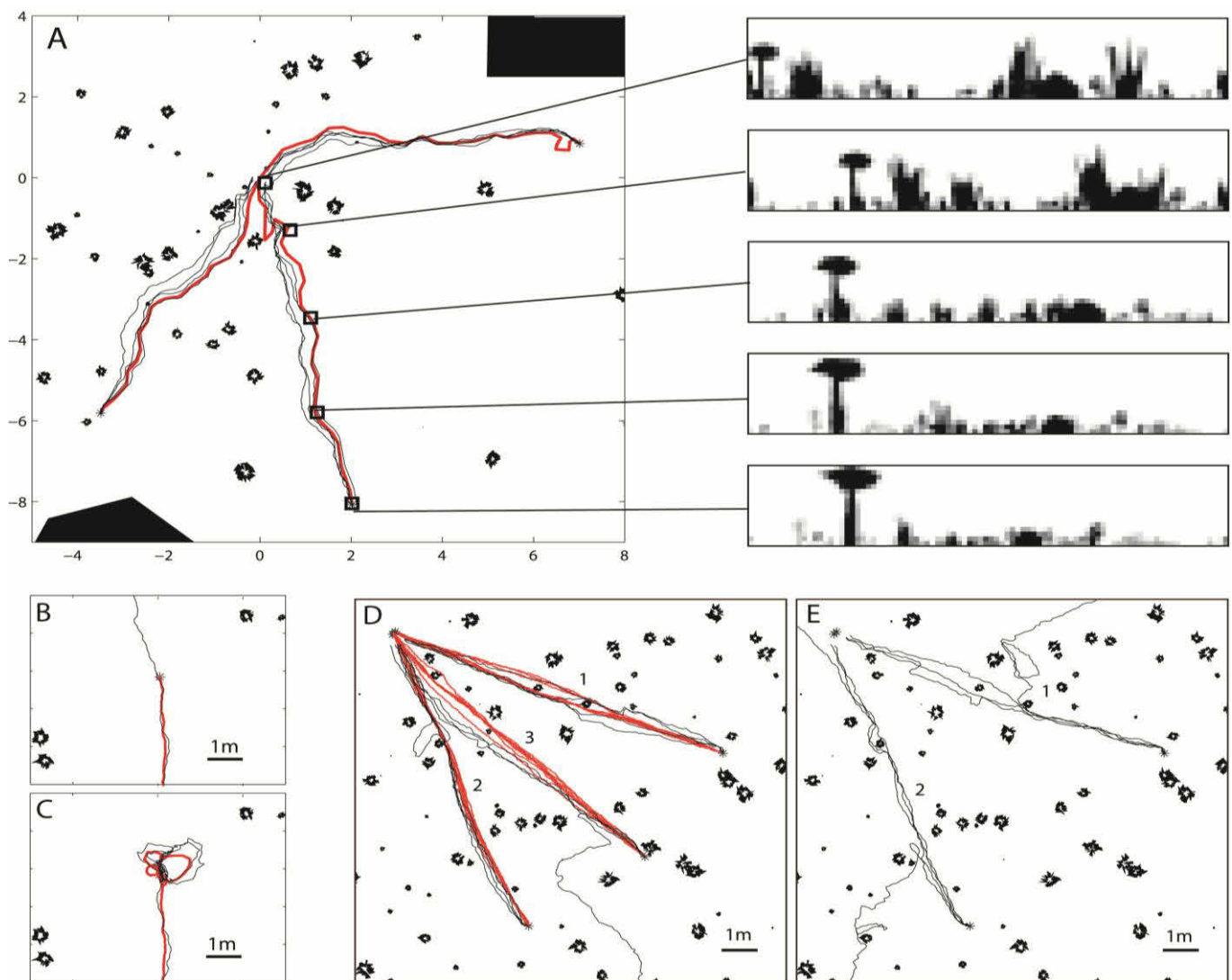


Figure 1: Route navigation with an embodied holistic model. The simulated world is viewed from above and is comprised mainly of small tussocks and a few larger more distant objects (trees and bushes). In all panels, red lines are training paths, black lines recapitulations. **A:** Successful return paths for three different routes. The panels to the right show example views covering  $360^\circ \times 68^\circ$  with  $4^\circ$  acuity from points along the training route (squares). **B-C:** Including learning walks prevents return paths from overshooting the goal. **B)** Without a learning walk the simulated ant overshoots and carries on in the direction it was heading as it approached the nest location. **C)** By including the views experienced during a learning walk the simulated ant, instead of overshooting, gets repeatedly drawn back to the location of the nest. **D-E:** Learning multiple routes. **D)** Route recapitulation performance (black lines) for each of three routes (red lines) that are learned with the same network. Testing of each of the routes is performed immediately following training on that route and prior to experience of other routes. Numbers by training routes show order in which routes were learnt. **E)** Performance on first two routes following learning of all routes, indicating that the route knowledge gained during the first two phases of learning is retained. Having learnt all 3 routes the network encodes 30m of route information.

## References

- Baddeley, B., Graham, P., Husbands, P. and Philippides, A. (2012) A Model of Ant Route Navigation Driven by Scene Familiarity. *PLoS Comput Biol*, 8(1): e1002336
- Cartwright, B. A. and Collett, T. S. (1983). Landmark learning in bees. *J Comp Physiol A*, 151: 521-543.
- Chittka, L. and Skorupski, P. (2011). Information processing in miniature brains. *Proc R Soc Lond B*, 278: 885-888.
- Möller, R. and Vardy, A. (2006) Local visual homing by matched-filter descent in image distances. *Biol Cybern*, 95: 413-430.
- Shettleworth, S. J. (2010). *Cognition, Evolution, and Behavior*, 2nd Edition. New York: Oxford University Press.
- Smith, L., Philippides, A., Graham, P., Baddeley, B. and Husbands, P. (2007). Linked local navigation for visual route guidance. *Adapt Behav*, 15: 257-271.
- Wehner, R. (2009). The architecture of the desert ant's navigational toolkit (Hymenoptera: Formicidae). *Myrmecol News*, 12: 85-96.

## From Microbiology to Microcontrollers: Robot Search Patterns Inspired by T Cell Movement

G. Matthew Fricke<sup>1</sup>, François Asperti-Boursin<sup>2</sup>, Joshua P. Hecker<sup>1</sup>, Judy L. Cannon<sup>2,3</sup> and Melanie E. Moses<sup>1,4,5</sup>

<sup>1</sup>Department of Computer Science, <sup>2</sup>Department of Molecular Genetics and Microbiology, <sup>3</sup>Department of Pathology,

<sup>4</sup>Department of Biology, University of New Mexico, Albuquerque, NM 87131

<sup>5</sup>Santa Fe Institute, Santa Fe, New Mexico 87501

mfricke@cs.unm.edu

### Abstract

In order to trigger an adaptive immune response, T cells move through lymph nodes searching for dendritic cells that carry antigens indicative of infection. We observe T cell movement in lymph nodes and implement those movement patterns as a search strategy in a team of simulated robots. We find that the distribution of step-sizes taken by T cells are best described by heavy-tailed (Lévy-like) distributions. Such distributions are characterized by many small steps and rare large steps. Our simulations show that heavy-tailed motion leads to dramatically faster search compared to Brownian motion, both in groups of T cells and in teams of robots. The mechanisms that cause heavy-tailed movement patterns in T cells are not fully understood. However, in robot simulations we find that heavy-tailed movement improves search speed whether that movement is caused by rules intrinsic to the robots or by adaptive response to extrinsic factors in the environment.

### Introduction

Biologically-inspired computation has a long history. Neural networks, genetic algorithms, and cellular automata are just a few well-known examples. In this study we observe immune cells searching within the three-dimensional space of mouse lymph nodes. We characterize T cell movement, demonstrate its effectiveness as a search strategy, and implement a similar search in simulated robots foraging for resources.

Robot teams can be used to perform real-world tasks such as surveying planetary surfaces and interplanetary space (Fink et al., 2005), land and sea mine clearance (Weber, 1995), pollution mapping by subsurface robots (Hu et al., 2011), and survivor location in hazardous environments (Birk and Carpin, 2006). The success of robot teams searching for resources in an unknown environment depends on the efficiency of the random search strategy employed.

### Biological Context

In order to mount an effective immune response, T cells must be activated in lymph nodes (Fig. 1). Activation occurs when a T cell discovers and interacts with a dendritic cell (DC) presenting a specific antigen. Antigens are markers that identify particular pathogens. Each T cell matches

a particular range of antigens. A DC presenting an antigen indicates that the corresponding pathogen has been encountered in the organism's tissues. If a T cell encounters a DC displaying cognate antigen then an immune response is triggered (Mackay et al., 2000).

To facilitate T cell activation, T cells and DCs interact within the T cell zone of lymph nodes (Fig. 1). The T cell zone is on the order of  $1\text{ mm}^3$  in the inguinal mouse lymph nodes we analyse. T cells and DCs are on the order of  $10\text{ }\mu\text{m}$  in diameter, so for each lymph node the T cell searches a space some  $10^8$  times its own volume. In secondary lymphoid organs, DCs usually comprise between 1% and 5% of the T cell zone's total cell population. Each T cell interacts with as many DCs as possible in order to maximize the probability of detecting a matching antigen (Mirsky et al., 2011). This imposes the need for efficient random search to mount an immune response.

Early response to infection depends on the rate at which DCs are discovered by T cells in the lymph node. The adaptive immune system is in an evolutionary arms race against an exponentially-growing pathogen population. That evolutionary pressure selects for efficient detection of, and response to, infection (Hedrick, 2004). Therefore we hypothesise that evolutionary pressure has produced an efficient mechanism for bringing T cells and DCs together, providing a model that can be used for random robotic search.

In this study, we use concepts derived from analysis of T cell search within lymph nodes to inform random robotic search. We identify the type of search used by T cells, then apply the observed three-dimensional search characteristics to a simple continuous space model. We simulate and characterize the performance of T cell inspired search strategies in robots using the iAnt robot system (Hecker et al., 2013). We found heavy-tailed search to be so effective for our simulated iAnts that we have begun incorporating it into the current multi-robot foraging algorithm. We find that T cells use a heavy tailed Lévy search and we show that this search strategy is more efficient than normally distributed random search.

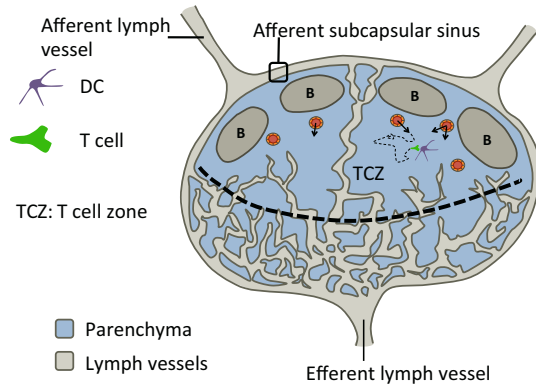


Figure 1: Diagram of a lymph node where T cells search for DCs.

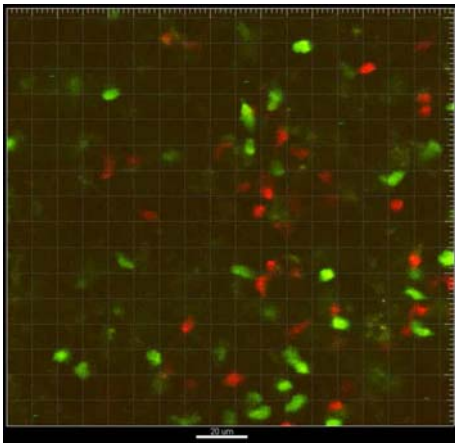


Figure 2: Image frame taken through two-photon microscopy of T cells (red and green) moving in the lymph node. The scale bar at the bottom of the image is 20 μm.

| Distribution | Formula                                                                      | Parameters            |
|--------------|------------------------------------------------------------------------------|-----------------------|
| Normal       | $\frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}$                 | $\sigma > 0, \mu > 0$ |
| Exponential  | $\frac{1}{\mu}e^{-\frac{x}{\mu}}$                                            | $\mu$                 |
| Log-normal   | $\frac{e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}}{x\sigma\sqrt{2\pi}}$          | $\sigma > 0, \mu$     |
| Gamma        | $\frac{1}{b^a\Gamma(a)}x^{a-1}e^{-\frac{x}{b}}$                              | $a, b$                |
| Power-law    | $\frac{1}{\sigma}\left(1 - k\frac{x-\theta}{\sigma}\right)^{-1-\frac{1}{k}}$ | $k, \sigma, \theta$   |

Table 1: Definitions and parameters for the five probability density functions used to model random motion.

## Stochastic Search

Search in 2- or 3-dimensional space is a common task in biological and engineered systems. Deterministic search strategies may be effective in relatively fixed environments where the distribution of search targets is known *a priori*. However, in environments where target distributions are unknown or change over time randomized search strategies are more effective (Stephens and Krebs, 1986; Acar et al., 2003).

Brownian motion is a common model of random walks. The turning angle between each step is drawn from a uniform distribution (Table 1, row 1).

Viswanathan et al. (2002), among others, described Lévy walks as a model for random walks that differs from classical Brownian motion. In that formulation, step lengths are drawn from a PDF over a power-law distribution. Power-law PDFs are scale-free and have heavy-tails with infinite variance. As a consequence, Lévy walks have many small steps and monotonically decreasing but non-zero probability of taking very large steps (or steps of any finite size). We use the Pareto (1895) formulation of the power law PDF (Table 1, row 5).

We define heavy-tailed distributions to be those with positive tails that approach zero less quickly than the exponentially-distributed PDF (the sub-exponential criteria, Bryson (1974)). Among many others, the log-normal (Table 1, row 3) and power-law distributions meet this criteria, whereas the normal and exponential distributions do not. We follow Shlesinger et al. (1999) in defining the heavy-tailed distribution of velocities (step lengths per time increment) as a *Lévy drive* and reserve Lévy walk for a heavy-tailed distribution of step-lengths with no time component.

Benhamou (2004) argues that Lévy walks are commonly misidentified and that the true picture is often of switching between travel phases from cluster to cluster and Brownian motion once a cluster is found. This pattern of search can be modelled by a correlated random walk (CRW) (Gillis, 1955;



Kareiva and Shigesada, 1983). CRWs are a class of random walks that incorporate non-uniform distributions in turning angle (Mårell et al., 2002).

### Related Robotics Work

Simulated robots have used Lévy walks in combination with chemotaxis-inspired gradient sensing (Nurzaman et al., 2009) and artificial potential fields (Sutantyo et al., 2010) to efficiently search unmapped spaces with range-limited sensors. In contrast we consider *free* Lévy drives in this study which do not allow long-range interactions between target and searcher.

In work related to our own, Van Dartel et al. (2004) evolved primitive neural controllers for agents foraging in a simulated world. They observed emergent Lévy walk patterns associated with increasing fitness, converging on parameters consistent with optimal foraging behaviour described by Viswanathan et al. (1999).

Harris et al. (2012) in their supplemental material describe computer simulation of Brownian motion and the generalized Lévy-walk search in a sphere. They report that the Lévy-walk was able to detect targets more efficiently than Brownian motion.

## Methods

### T Cell Observations

Lymph nodes were prepared according to the protocol described previously by Matheu et al. (2007). T cells were purified by nylon wool according to Allenspach et al. (2001) and labelled with one of two fluorescent dyes:  $1\text{ }\mu\text{M}$  (micromolar) Carboxyfluorescein diacetate succinimidyl ester (CFSE) or  $5\text{ }\mu\text{M}$  5-(and-6)-(((4-Chloromethyl) Benzoyl) Amino) Tetramethylrhodamine (CMTMR).  $5\text{ to }10 \times 10^6$  labelled T cells were injected intravenously into recipient mice. Fifteen to 18 hours later, after T cells migrated into lymph nodes, the inguinal lymph nodes were removed and recorded using two photon-imaging.

Imaging experiments were performed using a Biorad Radiance 2000 scanner mounted on an Olympus upright microscope with a chamber temperature of  $37^\circ\text{C}$ . Explanted lymph nodes were incubated with a  $37^\circ\text{C}$  solution of Dulbecco's Modified Eagle Medium (DMEM) bubbled with 95%  $\text{O}_2$  and 5%  $\text{CO}_2$  in order to preserve cell motility (Huang et al., 2007). T cell behaviour within a lymph node was monitored in the T cell area at a minimum of  $70\text{ }\mu\text{m}$  below the surface of the node. For 4D (3 spatial+1 time) analysis of T cell motility, multiple stacks in the z axis (z step =  $3\text{ }\mu\text{m}$ ) were acquired every 15-20 s (depending on the number of z stacks acquired) for 15-40 min, with an overall field thickness of  $40\text{--}60\text{ }\mu\text{m}$ .

Cell motility was analysed with Imaris 6.0 (Bitplane AG, Zurich, Switzerland). Tracks that lasted less than 3 time steps were removed from consideration. Tracks with total length or displacement from the start location less than

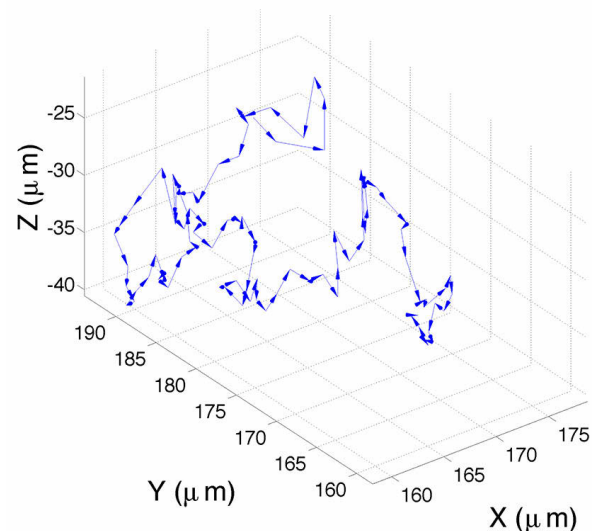


Figure 3: Example T cell track visualized from experiment data. Cell positions were captured every 14.93 seconds.

$17\text{ }\mu\text{m}$  over the course of the observation were assumed to be non-motile and discarded.

The point sequences generated by Imaris were used to create position vectors joining adjacent cell locations and the Euclidean norm for each vector was calculated. This provides a distribution of step sizes that were fit to probability distributions using Maximum Likelihood Estimation (MLE) described by Myung (2003).

The lab observations described were replicated seven times, resulting in 63,812 steps in 3,110 T cell tracks. The maximum velocity over all observations is  $1.9\text{ }\mu\text{m s}^{-1}$  with a mean of  $0.11\text{ }\mu\text{m s}^{-1}$ .

### Characterizing T Cell Search

Observed T cell population step sizes were fit to more than 50 PDFs. Of those distributions 5 PDFs were selected for further analysis: normal, log-normal, exponential, power-law and gamma. Harris et al. (2012) among others used normal and power-law PDFs to describe cell motion. Log-normal and exponential distributions are well known models of many biological processes. These four distributions form null hypotheses about the motion that we might expect to observe. The gamma probability density function is included because we found that, for several of the observations, it was the best model of the observed step-size distribution (Table 1).

In all cases, the bin sizes and binning methods were varied in order to reduce the effect of bin sizes on distribution fits. Adaptive binning rules described by Freedman and Diaconis (1981) were utilized along with various fixed bin sizes. Binning effects were not observed to be a factor in the fits.

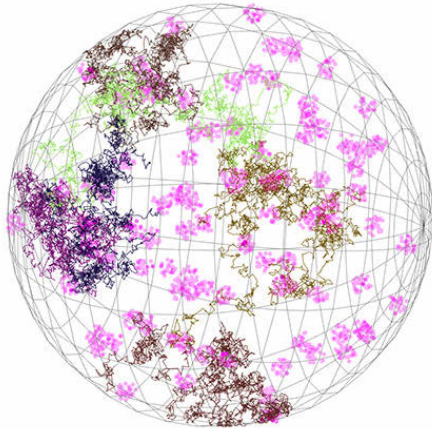


Figure 4: Clusters of targets with searcher tracks. Undiscovered DCs are pink, discovered DCs are red. Each coloured track corresponds to the path followed by each of the 6 searchers. 100 clusters of 30 targets are shown.

The relative *goodness of fit* (GoF) of each PDF to empirical data was evaluated using the Kolmogorov-Smirnov, the Bayesian information criterion (BIC), described in and the Akaike information criterion (AIC), as well as with log-likelihood measures (Table 2). Anderson-Darling calculates the integral of the area between the empirical data and the PDF. The Kolmogorov-Smirnov test matches the mean and variance of the observed data to the PDF and tests for normality. AIC and BIC incorporate the number of parameters available in the PDF to be fit to the observed distribution into the GoF value. AIC and BIC measure the information lost by replacing the observed data with a model.

Controversy exists around the identification of power-law PDFs and associated claims of Lévy walk observation. Many data sets that were not generated by a power-law PDF can be fit to a power-law distribution (Reynolds, 2008). We use techniques developed by Clauset et al. (2009) to address the fitting problems unique to power-law distributions. Distributions were fit to data and goodness of fit calculations were made in MATLAB and Statistics Toolbox Release 2013a (The MathWorks, Inc., Natick, Massachusetts, United States. 2013).

In order to determine whether the distribution of step sizes observed in the total population was due to the distribution of step sizes across tracks or within tracks we used the method of Petrovskii et al. (2011). Each track was scaled by the mean step length of that track and the distribution of the scaled tracks compared to the original tracks. Since the distribution was preserved after tracks were scaled the distribution is due to intra-track step lengths rather than differences in track means.

## Simulation in a 3D domain

Our 3D simulation models the search space as a continuous unit sphere (Fig. 4). For these experiments we used  $m = 16,384$  targets divided into  $n$  clusters, giving a target detection density on the same order as that estimated for DCs in lymph nodes.

Searchers were considered to have discovered a target if they came within a parameterized distance  $\gamma$  of a target. Targets were only available for discovery once. Search steps are treated as discrete in a continuous space, so detection of targets is checked at the end of each step and not at intermediary points. This detection radius encompasses the possible role of chemical gradients and DC dendrites reaching out into the surrounding space. Both mechanisms could be modelled by increasing  $\gamma$ .

Brownian motion is modelled as a sequence of fixed step lengths with uncorrelated turning angles. This results in motion that in the aggregate consists of movements along a trajectory (perhaps containing multiple steps) which is uncorrelated over any sub-sequence of trajectories. This formulation satisfies properties of Brownian motion identified by Einstein (1905): that trajectory lengths are uncorrelated, and displacement from a starting location tends towards a normal distribution. We tested this conclusion by repeating our experiments with Brownian motion modelled by step sizes drawn from a normal distribution and found the same performance.

Paths corresponding to the log-normal, exponential, gamma and power-law distributions were created using the same procedure, except that the scaling radial length was drawn from a PDF in which the mean value  $\mu$  is equal to  $r$ . This allows search to make relatively long jumps while making most of the jumps closer to the fixed step size of the discrete random walk. The simulation was written in C++ and PDFs calculated using the BOOST C++ Libraries 1.53.0 (Austern, 2005).

## iAnt Robot System

iAnt robots (Hecker et al., 2012) implement ant-inspired algorithms that mimic colonies of seed-harvester ants using a combination of individual memory and pheromone trail to collect resources and carry them to a central nest. Robots are equipped with ultrasound sensors, compasses, and cameras (Fig. 5) mounted on the robots which enable them to search for and find resources placed in various configurations. The iAnt simulator replicates the movement and sensing capabilities of these robots. iAnt behaviour has several phases, including a random search phase. The parameters for this search are determined by a genetic algorithm (GA) which evolves simulated iAnts and produces a strategy for the physical robots to use in the real resource collection task. Targets were distributed into 32 piles of 32 tags.

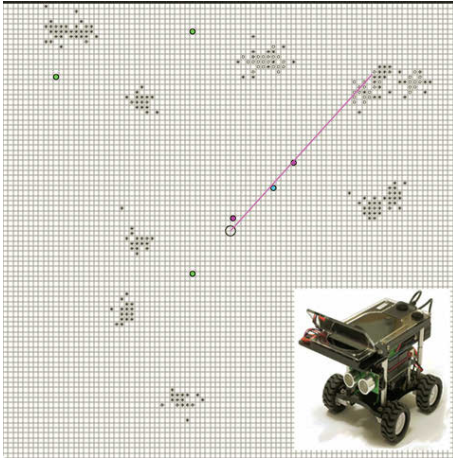


Figure 5: Simulation of the physical iAnt robots. Grey dots are QR tags which are the target of search. Circles are robot locations. Blue circles indicate a robot that has found a tag and is returning to a central location. Green circles are robots engaged in search. The pink line is a pheromone trail. Each of the 6 iAnts we have built is run by an on-board iPhone.

**Adaptive correlated random walk (ACRW)** In previous implementations, robots explored an experimental area using a random walk with fixed step size and a direction drawn from a normal distribution (Table 1, row 1). The standard deviation  $\sigma$  determines how correlated each step is with the previous step. In the ACRW  $\sigma$  varies depending on the observed density of targets in the search location. The search pattern therefore depends on the local density of targets observed by the robot.

We implemented five search strategies in simulation and compared them to one another: Brownian motion, a Lévy-like (log-normal) strategy, correlated log-normal, and two adaptive correlated random walks. The original ACRW used normally distributed step sizes; we compare that to an adaptive walk with log-normal distributed step sizes. With the exception of Brownian motion, each strategy has different parameters that are evolved by the iAnt genetic algorithm (GA). Log-normal search uses an evolved standard deviation to parameterize its log-normal step length distribution. Correlated log-normal search includes a second evolved standard deviation to parameterize a normal distribution of step angles. The adaptive correlated normal search has two evolved parameters that adapt step angle correlation, depending on whether robots have previously found resources or followed a pheromone trail. Adaptive correlated log-normal search uses the same two parameters to adapt step angle, as well as a third parameter to control the distribution of step lengths.

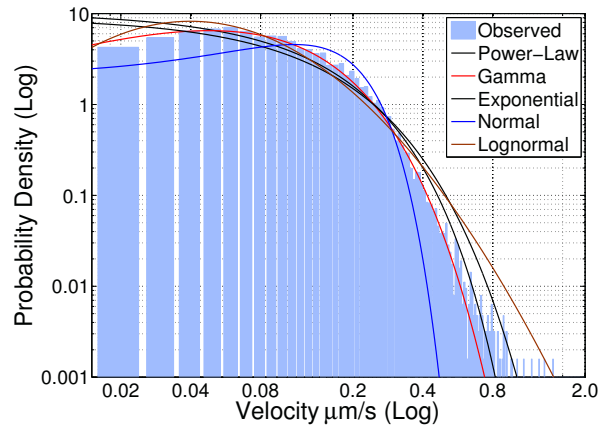


Figure 6: PDFs fit to a probability histogram of T cell velocities taken from all 7 experiments. Qualitatively and quantitatively the log-normal and gamma distributions fit the data more closely than the normal distribution. The normal distribution underpredicts how often large velocities occur, the log-normal distribution slightly underpredicts the number of small values, while the gamma distribution slightly underpredicts the number of large values.

| Distribution | AICc ( $\times 10^5$ ) | Log-likelihood ( $\times 10^4$ ) | K-S             | Relative AICc |
|--------------|------------------------|----------------------------------|-----------------|---------------|
| Normal       | -1.29                  | -6.49                            | 0.12, $p = 0$   | 0.84          |
| Exponential  | -1.52                  | -7.60                            | 0.1413, $p = 0$ | 0.94          |
| Log-normal   | -1.59                  | -7.95                            | 0.0748, $p = 0$ | 0.97          |
| Gamma        | -1.62                  | -8.11                            | 0.0460, $p = 0$ | 0.98          |
| Power-Law    | -1.64                  | -8.23                            | 0.0888, $p = 0$ | 1.0           |

Table 2: Goodness of fit using Akaike information criterion with finite size correction (AICc), Kolmogorov-Smirnov (K-S), and Log-likelihood tests.

## Results

### Characterizing T Cell Search

In all cases BIC and AIC measures were in agreement so only AIC results are presented.

We first asked what type of PDF best describes the type of T cell search occurring in lymph nodes. Table 2 shows the relative goodness of fit for each of the PDFs we considered when applied to the entire data set of T cell velocities. The K-S test rejects all the candidates as acceptable fits (small  $p$ -values), this is mostly due to the very large number of data points being fit. As the number of points increases, the tolerance for any deviation from the ideal analytic curve is reduced. Since empirical data necessarily differs from the ideal parametric PDF, with enough data points no distribution will be accepted as a fit to the data (i.e. fail to reject  $H_0$  that the observed data and proposed model come from the same PDF). As a result we use the AICc and log-likelihood methods to evaluate how well the distributions fit our data. Lower values for the three tests indicate better fits.

Considering the seven observational experiments sepa-

| Distribution | Parameters                                                                 |
|--------------|----------------------------------------------------------------------------|
| Normal       | mean ( $\mu$ ) = 0.11, sigma ( $\sigma$ ) = 0.08                           |
| Exponential  | mean ( $\mu$ ) = 0.1118                                                    |
| Log-normal   | log location ( $\mu$ ) = -2.5024, log scale( $\sigma$ ) = 0.84             |
| Gamma        | Shape (a) = 1.75, Scale (b) = 0.06                                         |
| Power-Law    | Shape (k) = -0.05, Scale ( $\sigma$ ) = 0.1, Threshold ( $\theta$ ) = 0.01 |

Table 3: Maximum likelihood parameter estimates.

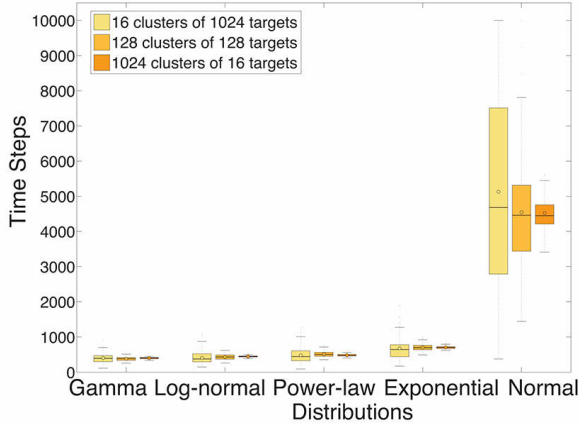


Figure 7: Comparison of search strategy performance across changes in target distribution. The y-axis is time for 6 searchers to find 1,000 targets. Bar = median, circle = mean.

rately resulted in power-law, log-normal and the gamma distributions being the top three fits, with one exception where the normal distribution ranked third in a single experiment. While all distributions are rejected by the Kolmogorov-Schmirnov test when fit to all the data, heavy-tailed distributions fit to individual experiments are not rejected. This leads us to believe that the data are not perfectly represented by any particular distribution, though our analysis shows that the heavy-tailed distributions are the best models to describe T cell movement in lymph nodes.

### 3D Performance

We then modelled search efficiency to test the search performance of heavy-tailed distributions and normal distributions using the model shown in Fig. 4. Evaluation of the search performance in simulation reinforces the functional similarity between the two heavy-tailed distributions (log-normal and power-law) and the gamma distribution which was parameterized to take on a heavy-tailed form. The two non-heavy-tailed search strategies (exponential and normal) failed to discover targets as quickly as the heavy-tailed search strategies (Fig. 7). Distribution parameters for the simulation were taken from those observed in PDFs fit to T cell motion (Table 3).

Heavy-tailed search did better than its competitors in finding targets quickly and with lower variance. Heavy-tailed search continues to find targets quickly when targets are

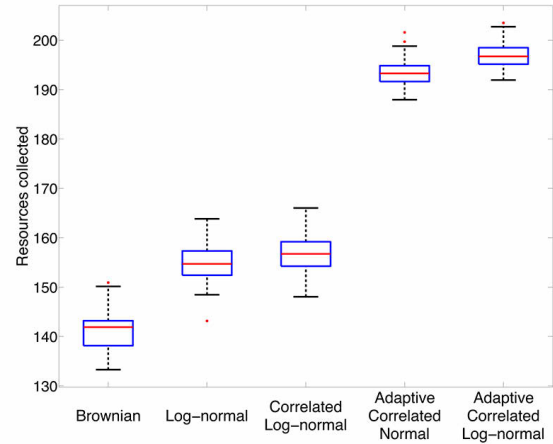


Figure 8: Comparison of search strategies in the iAnt simulator. We compare Brownian search, log-normal search, and adaptive correlated random walk strategies. While the heavy-tailed log-normal search performs better than Brownian search the correlated random walk is able to collect 25% more QR tags, and the adaptive correlated walks are able to collect 42% more tags in the same period.

highly clustered and separated by voids because they are able to cover gaps in less time than Brownian motion can. The cost heavy-tailed distributions pay is that they do not search the area they are in as exhaustively as Brownian motion. If a Brownian searcher happened to start near a cluster of targets it discovered many of those targets. If clusters were further removed from the searcher's initial placement then Brownian motion would have difficulty reaching the nearest cluster in a reasonable amount of time, this results in the high variance seen in Fig. 7. Heavy-tailed search is not as susceptible to the initial distribution because the rare but relatively large step sizes allow distances to be covered quickly so initial conditions have less impact on search success. Results presented are for 6 searchers but we observed similar behaviour for single searcher experiments.

### iAnt Performance

We then applied the heavy-tailed step size to the iANT robot simulation system for robotic target search. Performance of Brownian, log-normal, correlated log-normal, adaptive correlated Brownian and adaptive correlated log-normal search is shown in Fig. 8. The log-normal distribution was chosen to represent the heavy-tailed distributions due to simplicity of implementation and easy comparison to the normal distribution of Brownian motion.

The results for the 2D iAnt simulation were consistent with those from the 3D simulation. The heavy-tailed log-normal search outperformed Brownian motion in the iAnts. The adaptive strategies correspond to correlated random walks in which the correlation between step angles de-



depends on the target environment observed by the robots. Both adaptive search strategies significantly outperformed the non-adaptive algorithms. Robots using log-normal adaptive search collected more resources compared to robots using adaptive Brownian search. This difference is small but statistically significant:  $n = 198$ ,  $9.8\%$ ,  $p < 0.001$ .

The observed Lévy-walk pattern of step-sizes produced by the iAnt simulation performing the adaptive Brownian correlated walk are heavy-tailed.

## Discussion

We find that T cell movement in lymph nodes is better characterized by three different heavy-tailed distributions, power-law, log-normal and gamma than by exponential or Brownian motion (Fig. 6). Brownian and exponential distributions have commonly been used to model many processes in biology, economics, and physics (Mitzenmacher, 2004); however more recently, biological movement, including T cell movement through brain tissue (Harris et al., 2012), has been described by Lévy walks. Our results agree with previous studies showing that many biological systems adopt heavy tailed motion strategies, but do not follow an ideal power-law distribution of step sizes.

We have demonstrated that heavy-tailed distributions fit T cell motion in lymph nodes well, and are effective search strategies. Two of those heavy-tailed distributions are the log-normal and power-law distributions; Mitzenmacher (2004) discusses the history of debate across many fields and at many times regarding whether the lognormal or power-law distributions best model various phenomena. The heavy-tailed search strategies we simulated showed similar performance characteristics to one another. In that sense at least the particular distribution does not appear to matter as long as it allows a mixture of long steps with low likelihood and many small steps with high probability.

Our lymph node simulations show that the three heavy-tailed distributions search equally well, and much better than Brownian motion (Fig. 7). This is true of search for DCs distributed in a wide range of cluster densities. Thus the question of whether step size distributions are precisely power-law may not be relevant for determining the efficiency of the search process.

Viswanathan et al. (1999) shows that Lévy walks are optimal search when target clusters are sparse and targets are relatively slow compared to searchers. Numerous papers identified Lévy walks in biological data sets. Recently T cells have been found to perform Lévy walks in mouse brains as a response to parasitic infection (Harris et al. (2012)).

Using the iANT simulation, we found that walks that adapt to detected resources perform much better than simpler non-adaptive walks. Log-normal adaptive search performed only slightly better than Brownian adaptive search. However, we note that the observed distribution of steps sizes in the Brownian ACRW is also heavy-tailed. The com-

bination of step directions that are correlated over time (an intrinsic property of the robots), and adaptation to detected resources (an extrinsic property of the environment) results in an effective adaptive search.

In robots, the heavy tailed distributions caused by walks that adapt to environmental signals are much more effective than parametric heavy-tailed walks that are not adaptive. However adaptive walks may appear to have a heavy-tailed distribution of step lengths. It is currently unknown whether T cells might use the same adaptation strategy to respond to their environment within the lymph node. The lymph node has a complex structure, which includes the fibroblastic reticular cell (FRC) network (Chai et al., 2013). The FRC scaffold appears to serve multiple functions, but T cells have been observed, at times, to move along the FRC (Bajénoff et al., 2006). If the FRC network partially governs the motion of T cells, it would be considered an extrinsic factor as opposed to intrinsic cellular motion. Since T cells, and the lymph node in which they search for DCs, are here considered as part of the same system, we do not distinguish between intracellular and extracellular factors controlling the pattern of search. Therefore, even though extrinsic factors can shape T cell motion in vivo, our study doesn't discriminate whether T cells adapt their walks according to environmental cues in the lymph node. However this work suggests that as a hypothesis worth exploring.

In conclusion, our observations of T cell motion provided inspiration for efficient search in our robot systems and conversely observation of search in those robot systems has provided insight into the possible advantages that search pattern may provide to the immune system.

## References

- Acar, E. U., Choset, H., Zhang, Y., and Schervish, M. (2003). Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods. *The International Journal of Robotics Research*, 22(7-8):441–466.
- Allenspach, E. J., Cullinan, P., Tong, J., Tang, Q., Teschiaba, A. G., Cannon, J. L., Takahashi, S. M., Morgan, R., Burkhardt, J. K., and Sperling, A. I. (2001). Erm-dependent movement of cd43 defines a novel protein complex distal to the immunological synapse. *Immunity*, 15(5):739–750.
- Austern, M. (2005). Proposed draft technical report on c++ library extensions. *N1745*, pages 05–0005.
- Bajénoff, M., Egen, J. G., Koo, L. Y., Laugier, J. P., Brau, F., Glaichenhaus, N., and Germain, R. N. (2006). Stromal cell networks regulate lymphocyte entry, migration, and territoriality in lymph nodes. *Immunity*, 25(6):989–1001.
- Benhamou, S. (2004). How to reliably estimate the tortuosity of an animal's path:: straightness, sinuosity, or fractal dimension? *Journal of Theoretical Biology*, 229(2):209–220.
- Birk, A. and Carpin, S. (2006). Rescue robotics a crucial milestone on the road to autonomous systems. *Advanced Robotics*, 20(5):595–605.

- Bryson, M. C. (1974). Heavy-tailed distributions: properties and tests. *Technometrics*, 16(1):61–68.
- Chai, Q., Onder, L., Scandella, E., Gil-Cruz, C., Perez-Shibayama, C., Cupovic, J., Danuser, R., Sparwasser, T., Luther, S. A., Thiel, V., et al. (2013). Maturation of lymph node fibroblastic reticular cells from myofibroblastic precursors is critical for antiviral immunity. *Immunity*.
- Clauset, A., Shalizi, C. R., and Newman, M. E. (2009). Power-law distributions in empirical data. *SIAM review*, 51(4):661–703.
- Einstein, A. (1905). The theory of the brownian movement. *Ann. der Physik*, 17:549.
- Fink, W., Dohm, J. M., Tarbell, M. A., Hare, T. M., and Baker, V. R. (2005). Next-generation robotic planetary reconnaissance missions: a paradigm shift. *Planetary and Space Science*, 53(14):1419–1426.
- Freedman, D. and Diaconis, P. (1981). On the histogram as a density estimator: L 2 theory. *Probability theory and related fields*, 57(4):453–476.
- Gillis, J. (1955). Correlated random walk. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 51, pages 639–651. Cambridge Univ Press.
- Harris, T. H., Banigan, E. J., Christian, D. A., Konradt, C., Wojno, E. D. T., Norose, K., Wilson, E. H., John, B., Weninger, W., Luster, A. D., et al. (2012). Generalized lévy walks and the role of chemokines in migration of effector cd8+ t cells. *Nature*, 486(7404):545–548.
- Hecker, J. P., Letendre, K., Stolleis, K., Washington, D., and Moses, M. E. (2012). Formica ex machina: ant swarm foraging from physical to virtual and back again. In *Swarm Intelligence*, pages 252–259. Springer.
- Hecker, J. P., Stolleis, K., Swenson, B., Letendre, K., and Moses, M. E. (2013). Evolving Error Tolerance in Biologically-Inspired iAnt Robots. In *ECAL 2013 (in press)*.
- Hedrick, S. M. (2004). The acquired immune system-a vantage from beneath. *Immunity*, 21(5):607–616.
- Hu, H., Oyekan, J., and Gu, D. (2011). A school of robotic fish for pollution detection in port. *Biologically Inspired Robotics (Y. Liu and D. Sun, eds.)*, pages 85–104.
- Huang, J. H., Cárdenas-Navia, L. I., Caldwell, C. C., Plumb, T. J., Radu, C. G., Rocha, P. N., Wilder, T., Bromberg, J. S., Cronstein, B. N., Sitkovsky, M., et al. (2007). Requirements for t lymphocyte migration in explanted lymph nodes. *The Journal of Immunology*, 178(12):7747–7755.
- Kareiva, P. and Shigesada, N. (1983). Analyzing insect movement as a correlated random walk. *Oecologia*, 56(2-3):234–238.
- Mackay, I. R., Rosen, F. S., von Andrian, U. H., and Mackay, C. R. (2000). T-cell function and migrationtwo sides of the same coin. *New England Journal of Medicine*, 343(14):1020–1034.
- Mårell, A., Ball, J. P., and Hofgaard, A. (2002). Foraging and movement paths of female reindeer: insights from fractal analysis, correlated random walks, and lévy flights. *Canadian Journal of Zoology*, 80(5):854–865.
- Matheu, M. P., Parker, I., and Cahalan, M. D. (2007). Dissection and 2-photon imaging of peripheral lymph nodes in mice. *Journal of Visualized Experiments: JoVE*, (7).
- Mirsky, H. P., Miller, M. J., Linderman, J. J., and Kirschner, D. E. (2011). Systems biology approaches for understanding cellular mechanisms of immunity in lymph nodes during infection. *Journal of theoretical biology*, 287:160–170.
- Mitzenmacher, M. (2004). A brief history of generative models for power law and lognormal distributions. *Internet mathematics*, 1(2):226–251.
- Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1):90–100.
- Nurzaman, S. G., Matsumoto, Y., Nakamura, Y., Koizumi, S., and Ishiguro, H. (2009). Yuragi-based adaptive searching behavior in mobile robot: From bacterial chemotaxis to lévy walk. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 806–811. IEEE.
- Pareto, V. (1895). La legge della domanda. *Giornale degli economisti*, 10:59–68.
- Petrovskii, S., Mashanova, A., and Jansen, V. A. (2011). Variation in individual walking behavior creates the impression of a lévy flight. *Proceedings of the National Academy of Sciences*, 108(21):8704–8707.
- Reynolds, A. (2008). How many animals really do the lévy walk? comment. *Ecology*, pages 2347–2351.
- Shlesinger, M. F., Klafter, J., and Zumofen, G. (1999). Above, below and beyond brownian motion. *American Journal of Physics*, 67:1253.
- Stephens, D. W. and Krebs, J. R. (1986). *Foraging theory*. Princeton University Press.
- Sutantyo, D. K., Kernbach, S., Levi, P., and Nepomnyashchikh, V. A. (2010). Multi-robot searching algorithm using lévy flight and artificial potential field. In *Safety Security and Rescue Robotics (SSRR), 2010 IEEE International Workshop on*, pages 1–6. IEEE.
- Van Dartel, M., Postma, E., van den Herik, J., and de Croon, G. (2004). Macroscopic analysis of robot foraging behaviour. *Connection Science*, 16(3):169–181.
- Viswanathan, G., Bartumeus, F., Buldyrev, S., Catalan, J., Fulco, U., Havlin, S., Da Luz, M., Lyra, M., Raposo, E., and Eugene Stanley, H. (2002). Lévy flight random searches in biological phenomena. *Physica A: Statistical Mechanics and Its Applications*, 314(1):208–213.
- Viswanathan, G., Buldyrev, S., Havlin, S., Da Luz, M., Raposo, E., and Stanley, H. (1999). Optimizing the success of random searches. *Nature*, 401(6756):911–914.
- Weber, T. R. (1995). An analysis of lemmings: A swarming approach to mine countermeasures in the vsw/sz/bz. Technical report, DTIC Document.

# An evolutionary approach to road following: a simulated case study

Sebastian Clarke<sup>1</sup>, Frédéric Labrosse<sup>2</sup>, Vito Trianni<sup>3</sup> and Elio Tuci<sup>4</sup>

<sup>1,2,4</sup>Aberystwyth University, Llandinam Building, Aberystwyth, SY23 1RE, UK  
 CNR-ISTC, Via S. M. della Battaglia 44, 00185, Rome, Italy  
 sec14@aber.ac.uk<sup>1</sup>, flf@aber.ac.uk<sup>2</sup>, vito.trianni@istc.cnr.it<sup>3</sup>, elt7@aber.ac.uk<sup>4</sup>

## Abstract

Autonomous robot road following has been widely investigated since the early 1980s and, whilst much progress has been shown, there is still no system which displays 100% generality across all types of problem. This work shows a novel approach to the problem, using the methodology of Evolutionary Robotics to facilitate the autonomous emergence of flexible, robust and general behaviours. One of the unique aspects of this approach is to encourage the evolution of a dynamic strategy of colour perception: facilitating the combination of different channels of the colour space to perceive contrast across a range of scenes where this would otherwise be impossible. The results described herein demonstrate the capability of this methodology to produce controllers capable of generalising across a broad range of road shapes to which the agents have not been previously exposed. They also vindicate the effectiveness of a dynamic colour perception strategy, enabling the controllers to perceive contrast in a challenging variety of situations.

## Introduction

In the realm of autonomous vehicles, road following, i.e., the ability to detect and traverse a road surface without straying from the boundaries, is clearly an important problem. As such, it has received much attention from artificial intelligence and autonomous robotics researchers over the last thirty years (Dickmanns, 2002).

The earlier attempts employed hand-crafted controllers focusing on roads that are clearly demarcated and delineated (e.g., those that have either white lines or a clear high contrast delineation between the road and non-road surfaces, see Wallace et al., 1986; Waxman, 1985; Turk et al., 1988; Kuan et al., 1988; Dickinson and Davis, 1988). The general thrust of these approaches is to establish some sort of model based on the *a priori* assumption that the agent is already situated on the road, and by sampling its sensory information accordingly. This model is then maintained by monitoring prominent road features such as edges and lane demarcations. In order to improve the flexibility and the capability of a solution to follow more road types, later research looked at roads which have no clear demarcation, have amorphous or unclear delineation of the road edges

and have low contrast between the road and the background surface, and are subject to changing road conditions due to e.g., shadows or reflections (see Crisman and Thorpe, 1988; Kluge and Thorpe, 1990). To further address this problem, researchers started to investigate more adaptive, learning and connectionist based, rather than hand-crafted approaches (see Jochem et al., 1993; Pomerleau, 1997; Dickmanns, 2002). Recent success in this field has also been gained by combining these approaches with higher level, more complex models and reasoning, facilitated by the increase in available computing power over the last decade and spurred by high profile involvement from the military and commercial sectors (see Chen and Tsai, 1997; Aufrere et al., 2000; Urmson et al., 2008). Most recent approaches to the road following problem then, range from these complex, high-level models updated from multiple sensory sources and requiring significant computational power, to more simple, reactive and robust systems with lower model complexity and hence computational requirements (see Katramados et al., 2009; Ososinski and Labrosse, 2012).

The road following problem, in common with many others, has at its root a strong visual perception and feature extraction component. One of the problems in such visual discrimination tasks is a method of processing the input images in such a way as to reduce environmental distractions (in this case for example, those arising from shadows or reflections) so as to allow extraction of the features relevant to the problem, and to show clear contrast between the features to be extracted. One such technique could be to examine and combine different representational components of the image when transformed into colour spaces that separate luminosity (brightness) information from chrominance (colour) information. This can allow, for example, the efficient removal of shadows and effective dimensionality reduction of the problem by combining the remaining colour components in ratios that provide good performance over a range of possible scenes (see Woodland and Labrosse, 2005; Benedek and Szirányi, 2007; Finlayson et al., 2006).

However, it should be obvious that a fixed combination of colour components will never be able to show good con-

trast for all possible road following scenarios: we may well encounter scenes for which a broadly good combination of colour components actually shows very little contrast due to the nature of the colours represented in the scene. One solution to this problem may be to examine multiple components of the colour space simultaneously, dynamically choosing only those components (or combinations of components) that yield high contrast. However, for systems where input informational throughput may be limited such as low power embedded systems, or those using neural networks (i.e., where dimensionality reduction is necessary), it will not be possible to appraise all of this information concurrently. Similarly, there may be colour channels which yield very high contrast within the image, but not amongst the features that are relevant to the particular problem at hand. It should be clear then, that a method of dynamically combining various components of colour spaces in a way which aids feature detection by removing unwanted artefacts and increasing the contrast available between the relevant features for extraction is required for optimal performance of such visual perception and discrimination tasks.

With that in mind, we propose a solution to the road following problem using artificial neural network controllers synthesised by evolutionary computation techniques. We illustrate a system that is capable of successfully navigating roads in a variety of distinct simulated environments based on visual information in an integrated action-perception loop where the neural mechanisms that govern perception (identification of the road features) and action (motor activations and changes in colour perception) are not only tightly coupled, but are one and the same. In this study, we examine the combination of the standard RGB colour space components to increase contrast between the desired features, under the assumption that a similar process could be used with components of other colour spaces (that better separate luminance from chrominance) to augment this aim, and to reduce disturbances caused by shadow or reflections. We analyse how a system of dynamic colour perception contributes to the overall road-following effectiveness of the controllers and we show that without this adjunct to perception, the agent would be unable to solve all of the road following tasks presented in this simulated environment.

In summary then, the main objective of this study is to apply evolutionary robotics to the road following problem to produce a solution which will coordinate vision and action in a single, unified sensory-motor controller. Further, we aim to show that a dynamic method of dimensionality reduction, arrived at in tandem with the road following behaviour, will prove beneficial in solving the problem at hand. The evolved agents are capable of developing a general strategy for staying on poorly demarcated and delineated roads based on single camera visual input. Moreover, they are also able to dynamically adjust their colour perception, in real time, to increase the efficacy of road following in environments with



Figure 1: Pioneer robot

different or changing colour properties.

## The Robot and the Simulation Environment

A simulated robot is required to navigate various types of road using visual input. Our simulation models a Pioneer 3-AT 4-wheeled skid steer all terrain research robot as shown in Fig. 1. The simulation also comprises a 3D model of an environment, rendered using OpenGL (<http://www.opengl.org>) that provides the sensory information that the robot perceives through its camera. This environment contains only 3 visual components: a tiled textured horizontal plane on which the robot travels (the ground), a textured deviated surface rendered on this plane (the road), and a sky-box to provide the illusion of sky.

The virtual camera renders the 3D scene from the point of view of a camera mounted on top of the virtual robot. It is configured to have a frustum representative of real-world cameras that may be used to capture the scene. For final evolution to produce effective solutions, this would need to match exactly the specific camera being used. As this had not been chosen at the time of evolution, representative values were used with the understanding that changing the frustum would not significantly affect the results and that we are showing that the general concept of using camera images in this way is sound.

The visual input for the robot's controller is generated by significantly reducing the resolution of each camera image. Our simulated camera renders images at  $250 \times 200$  pixels. We overlaid a  $5 \times 5$  grid on each image, with each square grid covering 2000 pixels. The robot sensory input is made of 25 numerical values generated as follows. For each grid square, a mean value for each colour component is calculated by summing red, green, and blue components of each pixel within this square separately, then dividing by the total number of pixels residing in this grid square. In this way we have 3 values per grid square corresponding to the contributions of red, green and blue elements in the image. We then combine these components by multiplying each by  $\alpha$ ,  $\beta$ , and  $\gamma$  respectively, and then summing to produce a final single numeric value for each square.  $\alpha$ ,  $\beta$ , and  $\gamma$  are floats in  $[0, 1]$  generated by the robot controller at each time step. They are normalised so they sum to 1 hence they represent the ratios in which the red, green, and blue channels should be mixed



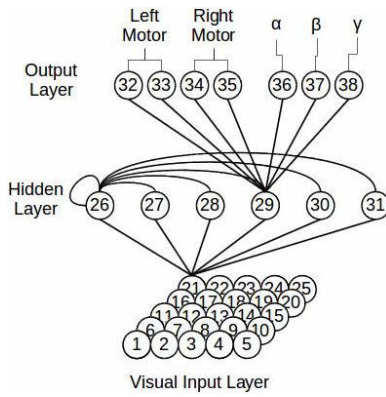


Figure 2: The neural network. The lines indicate the efferent connections for only one neuron of each layer. Each hidden neuron receives an afferent connection from each input neuron and from each hidden neuron, including a self-connection. Each output neuron receives an afferent connection from each hidden neuron.

respectively. As pixel colour values are in the range 0-255, we then divide by 255 to scale this value between 0 and 1.

### Controller and the Evolutionary Algorithm

The robot controller is composed of a continuous time recurrent neural network (CTRNN) of 25 visual input neurons, 6 inter-neurons, and 7 output neurons (see Beer and Gallagher, 1992). The structure of the network is shown in Fig. 2. The states of the output neurons are used to control the speed of the left and right wheels as explained later, and they define the ratios for colour mixing. The values of sensory, internal, and output neurons are updated using equations 1, 2, and 3.

$$y_i = gI_i; \text{ for } i \in \{1, \dots, 25\}; \quad (1)$$

$$\tau_i \dot{y}_i = -y_i + \sum_{j=1}^{31} \omega_{ji} \sigma(y_j + \beta_j); \text{ for } i = \{26, \dots, 31\}; \quad (2)$$

$$y_i = \sum_{j=12}^{15} \omega_{ji} \sigma(y_j + \beta_j); \text{ for } i = \{32, \dots, 38\}; \quad (3)$$

with  $\sigma(x) = (1 + e^{-x})^{-1}$ . In these equations, using terms derived from an analogy with real neurons,  $y_i$  represents the cell potential,  $\tau_i$  the decay constant,  $g$  is a gain factor,  $I_i$  with  $i = \{1, \dots, 11\}$  is the activation of the  $i^{th}$  sensor neuron,  $\omega_{ji}$  the strength of the synaptic connection from neuron  $j$  to neuron  $i$ ,  $\beta_j$  the bias term,  $\sigma(y_j + \beta_j)$  the firing rate (hereafter,  $f_i$ ). All sensory neurons share the same bias ( $\beta^I$ ), and the same holds for all motor neurons ( $\beta^O$ ).  $\tau_i$  and  $\beta_i$  with  $i = \{26, \dots, 31\}$ ,  $\beta^I$ ,  $\beta^O$ , all the network connection weights  $\omega_{ij}$ , and  $g$  are genetically specified networks' parameters. At each time step, the output of the left motor is  $M^L = f_{33} - f_{32}$ , and the right motor is  $M^R = f_{35} - f_{34}$ , with  $M_L, M_R \in [-1, 1]$ . The firing rates  $f_{36}, f_{37}, f_{38}$  are



Figure 3: Road tile construction and circle approximation

normalised such that  $\alpha + \beta + \gamma = 1$ . Cell potentials are set to 0 when the network is initialised or reset, and equation 2 is integrated using the forward Euler method with an integration time step  $\Delta T = 0.2$ .

A simple evolutionary algorithm using linear ranking is employed to set the parameters of the networks (Goldberg, 1989). The population contains 60 genotypes. Generations following the first one are produced by a combination of selection with elitism, recombination, and mutation. For each new generation, the three highest scoring individuals (“the elite”) from the previous generation are retained unchanged. The remainder of the new population is generated by fitness-proportional selection from the 30 best individuals of the old population. Each genotype is a vector comprising 243 real values (228 connections, 6 decay constants, 8 bias terms, and a gain factor). Initially, a random population of vectors is generated by initialising each component of each genotype to values chosen uniformly random from the range [0,1]. New genotypes, except “the elite”, are produced by applying recombination and mutation. Each new genotype has a 0.3 probability of being created by combining the genetic material of two parents. During recombination, one crossover point is selected. Genes from the beginning of the genotype to the crossover point are copied from one parent, the other genes are copied from the second parent. Mutation entails that a random Gaussian offset is applied to each real-valued vector component encoded in the genotype, with a probability of 0.05. The mean of the Gaussian is 0, and its standard deviation is 0.1. All vector component values are constrained to remain within the range [0,1].

### The visual scene

Textures that represent real world scenarios were chosen for the road and ground surfaces from the plethora available at the multitude of free texture resources on the internet. In order to simulate roads with amorphous nondescript edges, the edges of road textures were manually faded out using noisy paintbrush tools in image manipulation software, and then alpha-blended with the underlying ground texture.

We devised three complementary scenes, each of which featured only two of the three colour components (red, green and blue). The third component is randomly varied with noise, and hence unable to contribute to the final contrast between road and background visible in the scene. Giving

this randomly varied colour component strong weight in the colour combination would actively detract from the structured contrast visible between road and non road surfaces. In this way, the agent has to choose—by appropriately setting  $\alpha$ ,  $\beta$ , and  $\gamma$  for each environment—the colour components which assist in solving the problem, and disregard those that do not show the pattern that is being sought. This random colour variation was first undertaken at the pixel level, but later reimplemented to occur at the grid average level. In this way, with any static (i.e., not changing with respect to the environmental features) selection of colour combination there would always be one scene in which the controller was unable to detect any contrast. In addition, as there was a requirement that the agents evolve a general strategy that was able to cope with the road surface being both brighter (having higher values) than the background, and vice versa, scenes were carefully devised with the properties shown in Table 1. As long as the differences between the higher and lower values were kept constant, this should result in the following attributes. In 3 colour grey scale, all of the scenes would show no contrast between the road and non road surfaces. By choosing a single colour component, the agent would see positive contrast for the road in one scene, negative in another, and no contrast in the third. By choosing to combine two fixed colour components, the results would be the same. With this configuration, the only means by which the agent can successfully navigate all three scenes, and hence score maximum fitness, is by changing which colour components are examined between or during trials.

Each agent is evaluated against 6 different environments: two road shapes (one starting with a left bend, the other with a right bend) for each of the 3 colour scenes. At each generation, 6 different road shapes are generated using the following algorithm. First, a single square road tile is placed at the centre point of the ground plane at a world relative heading of  $0^\circ$ . A random angle is then chosen between two bounds (initially  $\pm 20^\circ$ ) and a centre point coordinate position is calculated for the new tile by applying basic trigonometry to translate it forward by 0.75 times the size of the road tile along this new angle. The new tile is then placed at this position and rotated by the angle, as shown in 3. This process is repeated for the number of road tiles required (in our experiment, 20) with the centre point position and total angle of each road tile stored for later use. In rendering, only the

angles are required: each road tile is rotated by, and translated along, this angle by OpenGL.

## The Fitness Function

All of the individuals in a population are evaluated against the same 6 environments to yield a proper comparison of the agent's performance, and new random road shapes were generated for each generation to expose them to as wide a variety of road following scenarios as possible. At the beginning of each trial ( $e$ ), the robots are placed at the start of the road at a random orientation between  $\pm 30^\circ$ .

The fitness function used in this approach is heavily inspired by that found in (Suzuki et al., 2005). This is a function which rewards forward progress of the robot, and is calculated and tallied at each time step based on the left ( $S_l$ ) and right ( $S_r$ ) wheel speeds, and the naïve straight-line distance reward applied at the end of the trial. The distance from the starting point to the final agent position is calculated as a percentage of the distance from the starting point to the end point of the road. This is multiplied by a reward factor and added to the fitness to further encourage the agent to reach the end of the road. There are situations however where this strategy will be counter-productive: e.g., when a road curves round so that the end point is closer to the start point than other positions on the road where the trial might terminate. However, it was concluded that with our random strategy of road building, such occurrences are rare enough not to adversely affect the system.

To combat the behaviour of a robot travelling in tight circles at the start point in order to “game” the fitness function without traversing any of the road, a penalty was added, such that the final fitness is halved if the agent remains on the starting road tile at the end of the trial. Furthermore, to encourage even more strongly the robot to avoid leaving the confines of the road, the fitness is multiplied by a factor of 1.2 if the trial was not terminated by a failed road bounds check. This serves to make a clear distinction between behaviours where the robot gets most of the way down the road, but leaves the road at the very end of the trial, and where the robot reaches the end of the road without doing so. Therefore, the final fitness ( $F$ ) for each genotype is calculated as:

$$F = \frac{1}{E} \sum_{e=0}^E f_e; \quad (4)$$

$$f_e = \eta \lambda \left( \frac{1}{T} \sum_{t=0}^{T'} g(S_l, S_r, t) + H \frac{D_a}{D_r} \right); \quad (5)$$

$$g(S_l, S_r, t) = (S_l^t + S_r^t) \left( 1 - \sqrt{\frac{|S_l^t - S_r^t|}{2 \times S_{max}}} \right); \quad (6)$$

where  $E = 6$  is the number of evaluations or trials per genotype,  $\eta = 0.5$  if the robot remains on the first road tile at the

Table 1: Table showing road scene properties. H refers to higher values, L refers to lower values.

|         | Red       | Green     | Blue      |
|---------|-----------|-----------|-----------|
| Scene 1 | Noise     | L on road | H on road |
| Scene 2 | H on road | Noise     | L on road |
| Scene 3 | L on road | H on road | Noise     |

end of the trial, otherwise  $\eta = 1.0$ ;  $\lambda = 1.2$  if the robot does not leave the road during the trial, otherwise  $\lambda = 1.0$ ;  $T$  is the maximum number of time steps in a trial (180 for these experiments);  $T'$  is the number of time steps experienced by the agent during this trial (for example, 100 where the trial is terminated after 100 time steps due to the robot leaving the road);  $D_a$  is the straight line distance from the start point to the final agent position at the end of the trial and  $D_r$  is the straight line distance from the start point to the centre of the final road tile;  $H = 1000$  is a constant,  $S_{max}$  being the maximum allowable wheel speed setting, in this case 300.

It should be noted that this fitness function makes no mention of the road, and the requirement to stay within its bounds. Rather, the trial is stopped prematurely when a robot leaves the road surface, so maximum fitness is only available to those agents that are capable of staying on the road throughout the full duration of the trial. To check whether a robot is on the road, we used the following criteria. The road is treated as a series of overlapping circles rather than squares, as shown in Fig. 3. By checking if the centre of the robot is further away from a centre point than the circle approximation radius, we can tell if it lies inside or outside of this circle.

## Results

Our objective is to synthesise controllers for autonomous robots required to visually navigate road surfaces without straying from the boundaries. We looked at roads which have unclear delineation of the edges, and situated in environments with different colour properties. The robots are required to dynamically combine—by appropriately setting  $\alpha$ ,  $\beta$ , and  $\gamma$  for each environment— various components of the colour spaces in order to detect road edges and to distinguish the road from the background.

Ten evolutionary runs, each using a different random initialisation were carried out for 2000 generations. Two evolutionary runs managed to generate robots with sufficiently high fitness to indicate that they are capable of successfully navigating all the three road scenes. The other eight runs produced only sub-optimal solutions. Due to the nature of the evolutionary process and fitness function, we cannot guarantee that the individuals with the highest fitness are those that have evolved the most robust general strategies. It is likely, in fact, that these fitness values actually represent the agents that are the “luckiest” with respect to the random variation occurring in the simulation. To deduce which are actually the most useful evolved individuals, we further evaluated a selection of the most fit genotypes across a broader range of tests with systematic variation.

### Post-evaluation test I

In this test, a suite of road following scenarios is generated, with parameters pertaining to road shapes and starting angles systematically adjusted between them. The fittest in-

dividual from the 100 fittest generations from each of the two successful evolutionary runs are then evaluated against the same set of conditions, allowing for a side by side comparison of their general effectiveness at such road following tasks. Together, the evaluations performed in Test I represent an examination of the effectiveness of the agent’s road following behaviour across a set of scenarios to which it has not been exposed during evolution. This test, therefore, is performed with the aim of demonstrating the generality of the road following solution produced.

The evaluation scenarios are produced by varying the allowable bounds between which angles ( $\theta$ ) are chosen for road tile placement. Two roads are generated for each of the following four configurations (for a total of 8 scenarios), where  $\theta$  is a randomly selected angle that each tile is placed at, between the following bounds: 1)  $\pm 20^\circ$ , 2)  $\pm 30^\circ$ , 3)  $\pm 40^\circ$ , and 4)  $\pm 50^\circ$ . 6 further scenarios are generated using roads featuring smooth, contiguous bend, where the tile placement angle is kept constant and uniform between road tiles. Three constant tile placement angle values are used to generate these roads, corresponding to shallow, medium and sharp corners:  $20^\circ$ ,  $30^\circ$  and  $40^\circ$ . Angles greater than these resulted in unrealistic looking roads with tighter corners than one would reasonably expect such a road-following vehicle to be capable of traversing. To avoid the road looping back on itself to form a circle, the direction of the placement angle is reversed once the total corner angle reaches  $110^\circ$ . Two roads are generated for each of the three cornering angles described (for a total of 6 scenarios), the first starting with a left turn, the second with a right. Finally, we included a straight road without corners to ensure that agents are effective on simpler tasks.

All of the above 15 evaluation scenarios (8+6+1) are generated with a fixed tile width of 140cm. Visually, they are of mostly uniform but slightly variable width due to the noisy fading to background of the road edges, representing more ill-defined roads. The visible road width is therefore roughly 120cm, or roughly twice the diameter of our simulated robot. Accordingly, the circle radius used for the road bounds approximation test is 60cm. Each road has a total traversable length of 28m. Each of the above road shapes is rendered in the three colour and texture combinations used for evolution, resulting in 45 evaluation scenes. Individuals are evaluated against each scene 5 times with different initial robot headings. Relative to the first road tile placement angle, these are:  $-45^\circ$ ,  $-22.5^\circ$ ,  $0^\circ$ ,  $22.5^\circ$  and  $45^\circ$ . A trial is considered successful if the robot successfully navigates to the penultimate road tile, to take into account erratic behaviour caused by the road ending in the robot’s field of view. In unsuccessful trials, the percentage of navigable road tiles successfully traversed is recorded.

The results of test I are shown in Table 2. We can see from this table that the best performers succeed in getting to the end of the road in almost 85% of the 225 individual evalua-

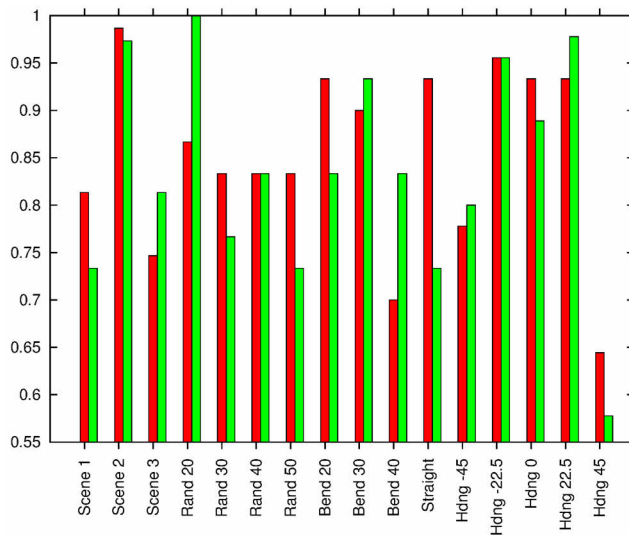


Figure 4: The performance of the best 2 individuals broken down by test type. Red bars for agent 2-1834, green bars for agent 6-1802.

tions. This is no mean feat, as they not only contain a challenging set of environments featuring diverse colours and textures and some obtuse, complex road shapes with sharp turns and currently unnavigable portions of the road in the field of view, but also place robots at a variety of starting headings, pointing more towards the sides of the road than the agents were evolved to cope with. It is also clear that both successful evolutionary runs produced roughly equally effective solutions.

Using the best individual from each run, measured by success percent, we looked at how they fare with the different problem types. In Fig. 4 the percentage success is broken down across the three coloured road scenes, the different angled random and bendy roads, the straight road, and different starting headings. Comparing the three coloured scenes

from Fig. 4 we can see that both agents perform better on scene 2 than the others. This could be due to it being a slightly easier colour/texture combination than the others, with slightly more contrast visible between road and non-road areas, but it may also be down to random occurrences in the pattern of evolution. The agents likely evolve to solve a particular coloured scene first, before learning to change their colour perception—it might simply be that both agents happened to learn this scene first, and therefore had more “practice” completing it.

Comparing the performance across the set of random roads with different placement angles, we can see that the agent from run 2 (hereafter, Agent 2) showed roughly uniform performance of around 85% across all 4 road types. This serves to show that it has evolved a highly general, effective road following behaviour, with difference in success rates largely unaffected by the coarseness of random road angles. The difference in performance between Agent 2 and the agent from run 6 (hereafter, Agent 6), at the 20° random road is striking and somewhat surprising. One might expect both agents to perform very well on this road as it most closely resembles those that they were evolved against. We can also see that Agent 6’s performance drops off on the other angled roads, in contrast to Agent 2’s broadly uniform performance, suggesting that Agent 6 has specialised on the 20° road to the detriment of the more difficult roads, whilst Agent 2 represents the more general solution at these road types, though showing somewhat worse performance on the simpler challenges. Interestingly, this situation does appear to be reversed for straight roads, where Agent 2 significantly outperforms Agent 6.

We can also see that both agents perform uniformly in a very effective manner when started at angles in the range  $-22.5^\circ \leq \theta \leq 22.5^\circ$ , with a drop off in performance for both agents when placed at angles beyond this range. This can be explained by the fact that although, during evolution, all agents are placed at world relative heading of  $0^\circ$ , the random changes in road tile angles will subject them to situations where they are pointing up to  $20^\circ$  from the centre line of the road, and they have hence developed strategies to mitigate this situation. What is surprising is that both agents perform worse when the agent is facing far to the left ( $45^\circ$  starting angle) than when it is subjected to such angles in the opposite direction. There is no obvious reason for this from the attributes of the evolutionary trials, other than that, through random fluctuations, the agents may have been exposed to more left turns than right. In summary then, the agents complete these evaluation tests with high effectiveness and there is not a great deal to choose between them. Any significant out-performance by one agent in a test, is made up for, either by more generality of the other agent across more tests, or an out-performance of the other agent on a different test. This demonstrates therefore, the capability of the evolutionary process to produce effective agents that encapsulate a

Table 2: The top ten Test 1 performers

| Run-Gen. | Success % | Mean distance | Std. dev. |
|----------|-----------|---------------|-----------|
| 2-1834   | 84.89     | 14.87         | 27.27     |
| 6-1802   | 84.00     | 5.25          | 12.89     |
| 2-1972   | 83.11     | 15.79         | 25.12     |
| 6-1939   | 83.11     | 8.92          | 15.27     |
| 2-1966   | 82.22     | 19.03         | 26.52     |
| 6-1825   | 82.22     | 15.28         | 20.16     |
| 2-1951   | 81.78     | 17.75         | 22.03     |
| 2-1838   | 81.33     | 13.10         | 25.68     |
| 6-1882   | 80.44     | 10.61         | 19.17     |
| 2-1808   | 80.44     | 7.95          | 16.80     |



general solution to the road following problem and are able to perform successfully across a wide variety of road types.

### Post-evaluation test II

Having analysed the generality and effectiveness of the road following behaviour in the previous test, it would also be useful to appraise the performance of the dynamic colour selection strategy evolved to show contrast in a variety of colour and texture scene combinations. To this end, the agent is exposed to 30 different scene texture combinations, representing all the possible combinations of the 6 textures shown in Fig. 5, using each texture as both road and ground surface. In order that the variance that we see in trial performance is due to the difference in texture combinations, rather than the varying performance across road shapes and starting angles, each agent is tested against roads generated using the same algorithm as that used in evolution to represent challenges of the sort with which the evolved agents should be most familiar. Similarly, rather than varying the starting heading of the agents on the road, each agent is started with a world relative heading of  $0^\circ$  as in evolution. 75 such roads are generated, and rendered in each of the 30 possible texture combinations, resulting in 2250 total evaluations per agent.

The results of this test, with respect to the different coloured scenes are plotted in Fig. 6. The labels for the bars of the histogram are in the form X/Y/Z where X is the agent number, Y is the first texture and Z is the second, according to the labels in Fig. 5. The bars are grouped by agent for each texture combination: the left bar being Agent 2, the right Agent 6, allowing direct comparison between the agents on different scenes. Each individual bar represents the mean percentage success rate for one agent across 150 trials: 75 road shapes with a particular road/ground texture combination, and the same 75 shapes with the reverse combination. The proportion of light and dark areas of each bar then represent the success rate for each of these individual reversals. For example, the bar labelled '2/B/T3' shows the performance of Agent 2, using a combination of 'Blue asphalt' and 'Asphalt 3' textures: the dark portion of the bar showing the success rate with 'Blue asphalt' road on 'Asphalt 3' ground, the light portion showing the reverse.

From the results of these tests, a few issues of note become apparent. Firstly it would appear that Agent 6 broadly outperforms Agent 2 in many of these trials. This likely suggests that Agent 6 has evolved a more general and effective

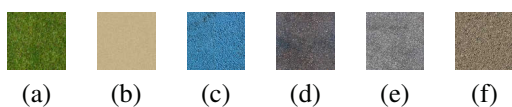


Figure 5: a) Grass (G), b) Sand (S); c) Blue asphalt (B); d) Asphalt 1 (T1); e) Asphalt 2 (T2); f) Asphalt 3 (T3)

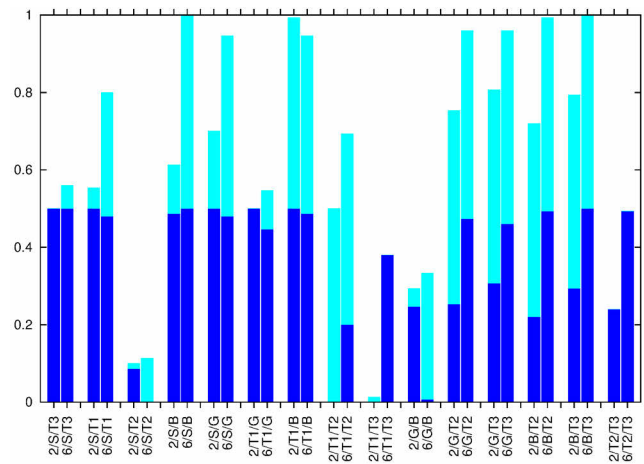


Figure 6: Comparison of both agent's performance across a range of scenes.

strategy for dynamic colour perception than Agent 2, though it is also possible that this is in some part due to it having evolved road following strategies that are more specialised to the type of road shapes present in evolution. We can also see that, for the majority of texture combinations, there is at least one reversal of road and ground textures that an agent is able to solve to a high degree of efficiency, i.e., there is normally a coloured bar portion at least 0.4 units long, a percentage success rate of 80 % for this road/ground texture combination. There is also a problem visible here though, as a number of scenes can only be solved effectively in one of the two reversals of road and ground texture. This shows a lack of generalisation in the solution. There is however another problem visible in this data. There are some scenes in which no agent can reliably detect contrast and solve the road, for either reversal of road and ground textures. This effect is most pronounced for S/T2 texture combination and it is believed this is due to the colours in both textures being too similar for either combination of them to reveal significant contrast.

### Conclusion

In this work, we have demonstrated a technique, using evolutionary robotics, to design effective road-following behaviours in simulated agents controlled by artificial neural networks. We have shown that, by presenting a set of challenges with diverse colour properties, we can encourage the evolution of an autonomous, dynamic approach to colour perception which enables evolved agents to perceive contrast in scenes where this would otherwise be impossible.

The process of evolution seems to have produced a range of effective and general solutions, which encapsulate not only a robust solution to road following, but also a system of dynamic colour perception that is able to show contrast be-

tween road and non-road surfaces across a range of scenes where a constant grey-scale conversion would fail. Through performing a range of tests, we ascertained which of the solutions is more generally effective, irrespective of their fitness scores which could be influenced by luck. By breaking down the results of post-evaluation tests, we have shown that the most effective agents show good generality in their road following ability, being capable of following roads differing significantly from those that they were evolved against. Their generality with respect to starting angles is not as strong, but this is expected as they were not deliberately exposed to a representative selection of these when evolving.

In examining the most effective produced agents with respect to their colour and road perception abilities it becomes clear that there are a few limits to their generality. In some instances (though not in others) the agents are not able to follow both reversals of road and ground textures, suggesting an inability to deal with certain combinations of values demarcating the road, even when contrast is visible. We have suggested that this could be mitigated either with more diverse evolution scenes, or an extra output node to reverse the visual input values. Similarly, we believe the slightly reduced performance on a couple of scenes in post-evaluation tests can also be improved with a better strategy to noise in the simulation. However, in spite of these slight problems, it seems the broad aim of evolving controllers with a dynamic approach to colour perception has been met, and the agents are able to detect contrast in a number of scenes where this would otherwise be impossible.

This work has been undertaken as a theoretical proof-of-concept: to show that the desired road following and dynamic colour perception behaviours can be produced through artificial evolution of neural network controlled robots. The transferral of such a system onto real robotics hardware has not been broached. We are aware that there are a number of issues which may affect the ability of this evolved controller to successfully cross the "reality gap". Future work will concentrate on this challenge.

## References

- Aufrere, R., Chapuis, R., and Chausse, F. (2000). A fast and robust vision based road following algorithm. In *Proc. of the 4<sup>th</sup> IEEE Intelligent Vehicles Symposium, 2000*, pages 192–197.
- Beer, R. D. and Gallagher, J. C. (1992). Evolving dynamic neural networks for adaptive behavior. *Adaptive Behavior*, 1(1):91–122.
- Benedek, C. and Szirányi, T. (2007). Study on color space selection for detecting cast shadows in video surveillance. *International Journal of Imaging Systems and Technology*.
- Chen, K.-H. and Tsai, W.-H. (1997). Vision-based autonomous land vehicle guidance in outdoor road environments using combined line and road following techniques. *Journal of Robotic Systems*, 14(10):711–728.
- Crisman, J. and Thorpe, C. (1988). Color vision for road following. In *Proc. of Conf. on Mobile Robots (SPIE)*, pages 1–10.
- Dickinson, S. and Davis, L. (1988). An expert vision system for autonomous land vehicle road following. In *Proc. of the Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 826–831.
- Dickmanns, E. (2002). The development of machine vision for road vehicles in the last decade. *Intelligent Vehicle Symposium, 2002. IEEE*, 1(1):268–281.
- Finlayson, G., Hordley, S., and Drew, M. (2006). Removing shadows from images. *Computer Vision ECCV 2002*, pages 1–14.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.
- Jochem, T., Pomerleau, D., and Thorpe, C. (1993). MANIAC: A next generation neurally based autonomous road follower. In *Proc. of Int. Conf. on Intelligence Autonomous Systems*, pages 15–18.
- Katramados, I., Crumpler, S., and Breckon, T. (2009). Real-time traversable surface detection by colour space fusion and temporal analysis. *Computer Vision Systems*, 5815(2007):265–274.
- Kluge, K. and Thorpe, C. (1990). Explicit models for robot road following. *Vision and Navigation*, pages 25–38.
- Kuan, D., Phipps, G., and Hsueh, A.-C. (1988). Autonomous robotic vehicle road following. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):648–658.
- Ososinski, M. and Labrosse, F. (2012). Real-time autonomous colour-based following of ill-defined roads. *LNAI: Advances in Autonomous Robotics*, pages 366–376.
- Pomerleau, D. (1997). Neural network vision for robot driving. *Intelligent Unmanned Ground Vehicles*, pages 1–22.
- Suzuki, M., Floreano, D., and Di Paolo, E. (2005). The contribution of active body movement to visual development in evolutionary robots. *Neural Networks*, 18(5-6):656–665.
- Turk, M., Morgenthaler, D., Gremban, K., and Marra, M. (1988). Vits-a vision system for autonomous land vehicle navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):342–361.
- Urmson, C., Anhalt, J., and Bagnell, D. (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field ...*, 25(February):425–466.
- Wallace, R., Matsuzaki, K., Goto, Y., Crisman, J., Webb, J., and Kanade, T. (1986). Progress in robot road-following. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 1615–1621.
- Waxman, A. (1985). Visual navigation of roadways. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 862–867.
- Woodland, A. and Labrosse, F. (2005). On the separation of luminance from colour in images. In *Proc. of the Int. Conf. on Vision Video and Graphics*. The Eurographics Association.

# Evolving Error Tolerance in Biologically-Inspired iAnt Robots

Joshua P. Hecker<sup>1</sup>, Karl Stolleis<sup>1</sup>, Bjorn Swenson<sup>1</sup>, Kenneth Letendre<sup>1,2</sup> and Melanie E. Moses<sup>1,2,3</sup>

<sup>1</sup>Department of Computer Science, University of New Mexico, Albuquerque, NM 87131

<sup>2</sup>Department of Biology, University of New Mexico, Albuquerque, NM 87131

<sup>3</sup>External Faculty, Santa Fe Institute, Santa Fe, NM 87501

{jhecker,melaniem}@cs.unm.edu {kstolleis,kletendr}@unm.edu

## Abstract

Evolutionary algorithms can adapt the behavior of individuals to maximize the fitness of cooperative multi-agent teams. We use a genetic algorithm (GA) to optimize behavior in a team of simulated robots that mimic foraging ants, then transfer the evolved behaviors into physical iAnt robots. We introduce positional and resource detection error models into our simulation to characterize the empirically-measured sensor error in our physical robots. Physical and simulated robots that live in a world with error and use parameters adapted specifically for an error-prone world perform better than robots in the same error-prone world using parameters adapted for an error-free world. Additionally, teams of robots in error-adapted simulations collect resources at the same rate as the physical robots. Our approach extends state-of-the-art biologically-inspired robotics, evolving high-level behaviors that are robust to sensor error and meaningful for phenotypic analysis. This work demonstrates the utility of employing evolutionary methods to optimize the performance of distributed robot teams in unknown environments.

## Introduction

Multi-agent simulations have been used to evolve behaviors which are then transferred into physical robots (Nelson et al., 2004; Singh and Parhi, 2011). Simulations rapidly generate multiple viable solutions, allowing researchers to test many possible scenarios and make informed decisions about which physical experiments to run. Such simulations should focus on physical fidelity by replicating the environment, hardware constraints, and sensor error of the real robots (Brooks, 1992).

A particularly challenging class of problems for multi-robot systems is central-place foraging (Matarić, 1994; Panait and Luke, 2004). For this task, robots are programmed to search an area for resources and aggregate these resources at a central location. Foraging is considered a canonical task for distributed robotics: foraging can be instantiated into a number of real-world applications such as hazardous waste clean-up (Parker, 1998), land mine detection and removal (Gage, 1995; Kong et al., 2006), search and rescue (Kitano et al., 1999), and extraplanetary exploration (Curtis et al., 2003; Tunstet et al., 2008). For applications

where the physical environment may vary over time and the distribution of resources is most likely unknown, evolutionary approaches allow robot teams to adapt their behavior to each particular scenario.

Our robots use a central-place foraging algorithm (CPFA) based on the foraging behavior of ants (Hecker et al., 2012; Hecker and Moses, 2013). The CPFA is parameterized by a GA in a multi-agent simulation which emulates the physical robot experiments. Our simulation evolves parameters in a parsimonious model of biological ant behavior, and our iAnt robots use these parameters to forage for resources in an experimental area. We investigate the effects of sensor error on physical and simulated robot performance. We demonstrate the utility of this approach by measuring the number of resources that robots collect using parameters adapted and not adapted to error.

## Previous Work

We conducted manipulative field studies on three species of *Pogonomyrmex* desert seed-harvester ants (Flanagan et al., 2012). Colonies were baited with dyed seeds distributed in a variety of pile sizes around each ant nest. We calculated foraging rates for each distribution and found that ants collected seeds faster when seeds were more clustered. Computer simulations used genetic algorithms to find individual ant behavioral parameters that maximized the seed collection rate of the colony. Simulated ants foraging with those parameters mimicked the increase of seed collection rate with the amount of clustering in the seed distribution when ant agents were able to remember and communicate seed locations (Paz Flanagan et al., 2011).

We also observed how individual parameters and overall fitness change with different distributions of resources and different numbers of simulated agents performing a central-place foraging task (Letendre and Moses, 2013). Parameters evolved for specific types of resource distributions were swapped and then fitness was measured for the new distribution; for example, parameters optimized for a clustered distribution were tested on random distributions of resources. Simulated agents incurred as much as a 50% decrease in fit-

ness when using parameters on a distribution different from the one for which they were optimized.

We then modified our multi-agent central-place foraging simulation to model the physical environment and hardware constraints of our iAnt robot platform (Hecker et al., 2012). We adapted our existing GA to evolve parameters for our iAnt robots. The evolved parameters were then transferred into the physical robots. Simulated teams collected three to four times as many resources as the real robot teams. We hypothesized that this discrepancy resulted from a *reality gap* between the error-free simulated world and the sensor error experienced by the physical robots.

Most recently, we incorporated a probabilistic error model into our multi-agent iAnt simulator in a workshop paper (Hecker and Moses, 2013). In this preliminary study, we added varying amounts of noise to agents' physical positions and their ability to detect resources, and analyzed the response of the genetic algorithm by observing individual behavioral parameters. We saw that increased positional error reduced resource collection, and induced the GA to select for a lower likelihood of returning to locations where resources were previously found. Increased detection error also reduced resource collection, as well as influencing the GA to select behaviors that searched local areas more thoroughly when only a few resources were detected. These behaviors indicated that the GA was able to evolve parameters appropriate to the sensor error used in the simulation.

We build on this prior work by *a)* simplifying the CPFA which improves performance and makes it easier to interpret why parameters are evolved to different values in different experiments; *b)* updating the iAnt simulator to more accurately reflect physical reality; *c)* testing the CPFA on new resource distributions; and *d)* implementing error-adapted parameters in experiments in physical robots.

## Background

Research in evolutionary robotics (ER) primarily focuses on using evolutionary methods to develop controllers for autonomous robots. Controllers can be evolved in simulation and subsequently transferred into physical robots (Nelson et al., 2004; Singh and Parhi, 2011), or evolved directly in real robots through embodied evolution (Watson et al., 2002). Following principles outlined by Brooks (1991), work in ER has focused on bridging the reality gap between simulated and real robots to improve the performance of evolved controllers in the physical world (Jakobi et al., 1995). Neural networks have been used in combination with evolutionary methods to evolve controllers for simulated robot agents with random sensor noise; controllers were subsequently transferred to real robots with varying degrees of success (Nolfi et al., 1993; Miglino et al., 1995; Jakobi, 1997).

State-of-the-art robotic simulators such as Stage (Vaughan, 2008) and ARGoS (Pinciroli et al., 2011) can

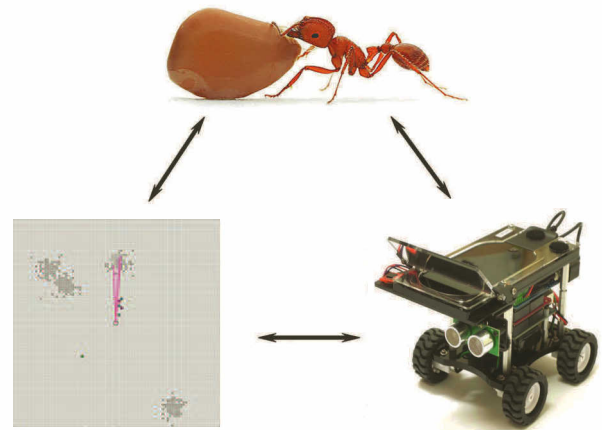


Figure 1: Our approach leverages studies on biological ants, multi-agent simulations guided by genetic algorithms, and our physical iAnt robot platform.

be used to model large robot teams with realistic, complex physical kinematics, but they do not incorporate any learning or evolutionary methods that allow simulated agents to adapt to unknown environments. Neither simulator includes sensor noise in its standard implementation, however Pinciroli et al. (2012) recently modified ARGoS to incorporate an actuator noise model, generating performance matching results from positional error observed in real robots.

Previous work on multi-robot group foraging tasks used reinforcement learning to train robots on higher-level behaviors, rather than lower-level motor controllers or basic directional responses (Matarić, 1997a,b). Robots learned when to switch between behaviors in a fixed repertoire set through positive and negative reinforcement related to their foraging success. We follow this high-level learning approach in the design of our CPFA.

Our approach (see Figure 1) differs from previous approaches in that we do not attempt to evolve basic primitive behaviors from the ground up. Instead, we model existing biological ant behaviors that have evolved naturally over millions of years. We use a genetic algorithm to parameterize these behaviors in our simulated agents, then we transfer those behaviors into physical robots. Evolved parameters control the sensitivity threshold for triggering behaviors, the likelihood of transitioning from one behavior to another, and the length of time each behavior should last.

We extend the state of the art in evolutionary and biologically-inspired robotics by *i)* evolving high-level behaviors that are *ii)* robust to real-world sensor error and *iii)* meaningful for phenotype-level analysis.

## Methods

We present our simulated model of ant behavior, detailed pseudocode and diagrams explaining our simplified CPFA,



probabilistic models of physical sensor error in the iAnt robot platform and implemented in our multi-agent system, and the design of our simulated and physical experiments.

### Ant Behavior Model

*Pogonomyrmex* seed-harvester ants follow a central-place foraging strategy to aggregate food at their colony's single nest. These foragers typically leave their nest, travel in a relatively straight line to some location on their territory, and then switch to a correlated random walk to search for seeds. A foraging ant who has located a seed brings it directly back to the nest. Foragers often return to a location where they have previously found a seed in a process called site fidelity (Moses, 2005; Beverly et al., 2009; Flanagan et al., 2012). Our recent work indicates that combining site fidelity with occasional laying of pheromone trails to dense piles of food may be an effective component of these ants' foraging strategies (Paz Flanagan et al., 2011; Letendre and Moses, 2013).

We incorporate key behaviors observed in our previous field studies on desert seed-harvester ants (Flanagan et al., 2012) into our multi-agent simulation and physical iAnt robots. We model probabilistic actions and state transitions using eight evolvable parameters, detailed in Table 1. These are simplifications of our earlier CPFA algorithm (Hecker et al., 2012). Modifications have been made since our most recent work in an effort to increase parsimony (Hecker and Moses, 2013), such as removing the parameter for probabilistically abandoning a pheromone waypoint:

- **State transitions:** Robots switch between two behaviors:
  - **Traveling:** In the absence of information, a robot at the nest will select a random direction and begin traveling. At each step of traveling, robots have a probability  $p_s$  of transitioning to search behavior.
  - **Searching:** At each step of searching, robots who have not found a resource have a probability  $p_t$  of returning to the nest.
- **Correlated random walk:** Robots explore regions using a random walk with a fixed step size and a direction  $\theta_t \sim \mathcal{N}(\theta_{t-1}, \sigma)$  at time  $t$ . The standard deviation  $\sigma$  determines how correlated the direction of the next step is with the direction of the previous step.  $\sigma$  depends on whether an agent has prior information through the use of site fidelity or pheromones:
  - **Uninformed search:** If an agent has not used site fidelity or pheromones, then  $\sigma = \omega$ .
  - **Informed search:** If an agent has arrived at a site by using site fidelity or pheromones, then  $\sigma = \omega + (4\pi - \omega) * e^{-\lambda_{id} * t}$ , where  $\sigma$  decays to  $\omega$  as time  $t$  increases.
- **Information:** Previous ant studies have demonstrated the ability of ants to count event frequencies in estimating

| Parameter      | Description                   | Initialization Function |
|----------------|-------------------------------|-------------------------|
| $p_t$          | Probability of traveling      | $\mathcal{U}(0, 1)$     |
| $p_s$          | Probability of searching      | $\mathcal{U}(0, 1)$     |
| $\omega$       | Uninformed search correlation | $\mathcal{U}(0, 4\pi)$  |
| $\lambda_{id}$ | Informed search decay         | $exp(5)$                |
| $\lambda_{lp}$ | Rate of laying pheromone      | $exp(1)$                |
| $\lambda_{fp}$ | Rate of following pheromone   | $exp(1)$                |
| $\lambda_{sf}$ | Rate of site fidelity         | $exp(1)$                |
| $\lambda_{pd}$ | Rate of pheromone decay       | $exp(10)$               |

Table 1: Set of 8 parameters evolved in simulation guided by genetic algorithms. At the start of a simulated run, parameters in each colony are initialized using randomly sampled values from their associated initialization function. The first 3 parameters are initially sampled from a uniform distribution, and the last 5 from exponential distributions within the stated bounds.

nest size (Mallon and Franks, 2000), travel distance (Witlinger et al., 2006), and encounter rates with other ants (Prabhakar et al., 2012). In our simulation, when an agent finds a resource, it stores a count  $c$  of additional resources in the 8-cell neighborhood of the found resource. This count  $c$  represents an estimate of the density of resources in the local region, and the agent uses  $c$  to decide when to use site fidelity, lay a pheromone waypoint, or follow a pheromone waypoint:

- **Site fidelity:** A robot returns to a previously found resource location if  $F_{sf}(c) > \mathcal{U}(0, 1)$ , where  $F_{sf}(x) = 1 - e^{-\lambda_{sf} * (x+1)}$ .
- **Laying pheromone:** A robot creates a pheromone waypoint for a previously found resource location if  $F_{lp}(c) > \mathcal{U}(0, 1)$ , where  $F_{lp}(x) = 1 - e^{-\lambda_{lp} * (x+1)}$ . New pheromone trails are initialized with a value of 1.
- **Following pheromone:** Upon returning to the nest, a robot follows a pheromone waypoint to a previously found resource location if  $F_{fp}(c) > \mathcal{U}(0, 1)$ , where  $F_{fp}(x) = 1 - e^{-\lambda_{fp} * (9-x)}$ . Waypoints are selected with probability proportional to their pheromone value.
- **Pheromone decay:** Pheromone waypoints decay exponentially over time  $t$  as  $e^{-\lambda_{pd} * t}$ . Waypoints are removed from the simulation once their value drops below a threshold of 0.001.

Four parameters that are of interest in our analysis are the informed search decay rate ( $\lambda_{id}$ ), the rate of using site fidelity ( $\lambda_{sf}$ ), the rate of laying pheromone ( $\lambda_{lp}$ ), and the rate of following pheromone ( $\lambda_{fp}$ ). Lower values of informed search decay ( $\lambda_{id}$ ) cause the robots to use a less correlated

random walk, and thus a more random and thorough local search, for a longer period of time when they have information pertaining to a high density of resources at a particular location.

In both simulated and physical robots, we simulate pheromone trail use by maintaining a list of waypoints. Pheromone strength of each waypoint evaporates over time ( $\lambda_{pd}$ ). Physical marking is not possible with real robots, and therefore our simulated agents follow the same protocol.

### Search Algorithm

CPFA pseudocode is shown in Algorithm 1. Note that probabilities of using site fidelity ( $F_{sf}(c)$ ), laying pheromone ( $F_{lp}(c)$ ), and following pheromone ( $F_{fp}(c)$ ) are generated using the equations discussed in the previous subsection. Figure 2(a) shows a state diagram of the algorithm, and Figure 2(b) illustrates an example of one possible cycle through the search behavior loop. An iAnt robot is shown in Figure 2(c).

---

#### Algorithm 1 Biologically-Inspired CPFA

---

```

Disperse from nest to random location
while experiment running do
  Conduct uninformed correlated random walk
  if resource found then
    Count number of resources  $c$  near current location  $l_f$ 
    Return to nest with resource
    if  $F_{lp}(c) > \mathcal{U}(0, 1)$  then
      Create pheromone waypoint for  $l_f$ 
      Pheromones followed by robots at nest
      Pheromones decay over time
    else
      if  $F_{sf}(c) > \mathcal{U}(0, 1)$  and  $F_{fp}(c) < \mathcal{U}(0, 1)$  then
        Return to  $l_f$ 
        Conduct informed correlated random walk
      else
        Check for pheromone
        if pheromone found and
           $F_{fp}(c) > \mathcal{U}(0, 1)$  and  $F_{sf}(c) < \mathcal{U}(0, 1)$  then
            Travel to pheromone location  $l_p$ 
            Conduct informed correlated random walk
          else
            Choose new random location
          end if
        end if
      end if
    end if
  end while

```

---

### Physical Sensor Error

Two sensing components are precise in simulation but error-prone in our physical iAnt robot platform: positional measurement and resource detection. Our physical robots use

a combination of ultrasonic distance, magnetic compass headings, time-based odometry, and an on-board forward-facing camera to estimate their position within the experimental area. Resource detection is accomplished using a downward-facing camera to read barcode-style QR tags.

We measured positional error in five physical robots while localizing to measure the absolute position of a found resource, and while traveling to a location informed by site fidelity or pheromones. We replicated each test 20 times per robot; means and standard deviations for both types of positional error were calculated using 120 samples each. For robots localizing at a true position of (0 cm, 0 cm), we observed a measured position of  $(-18 \pm 79 \text{ cm}, -15 \pm 47 \text{ cm})$ , whereas robots traveling to a true position of (0 cm, 0 cm) had a measured position of  $(1.6 \pm 45 \text{ cm}, 64 \pm 110 \text{ cm})$ .

Positional error is modeled by perturbing the physical position of an agent from  $(x, y)$  to  $(x', y')$ , such that  $x' \sim \mathcal{N}(x + \hat{x}, \sigma_x)$  and  $y' \sim \mathcal{N}(y + \hat{y}, \sigma_y)$ . That is,  $(x', y')$  is sampled from a normal distribution with mean equal to the true position  $(x, y)$  offset by  $(\hat{x}, \hat{y})$ , and standard deviation  $(\sigma_x, \sigma_y)$ . We impose this positional perturbation twice: once when a robot finds a resource, and again when a robot leaves the nest using site fidelity or following a pheromone waypoint to a known location.

We observed resource detection error for physical robots searching for resources, and for robots searching for neighboring resources. Resource-searching robots attempt to physically align with a QR tag, using small left and right rotations and forward and backward movements to center the tag in their down-facing camera. Robots searching for neighboring resources do not use this alignment strategy, but instead simply rotate  $360^\circ$ , scanning for a tag every  $10^\circ$  with their down-facing camera. We replicated each test 20 times for three different robots; means for both types of resource detection error were calculated using 60 samples each. We observed that resource-searching robots detected 55% of tags and neighbor-searching robots detected 43% of tags. Resource detection is modeled as a fixed probability  $d_r = 0.55$  for resource-searching robots, and  $d_n = 0.43$  for neighbor-searching robots.

### Experimental Design

Each experimental physical trial on a 100 m<sup>2</sup> concrete surface runs for 30 minutes. An illuminated beacon marks the center ‘nest’ to which the robots return once they have located a resource. This center point is used for localization and error correction by the robots’ ultrasonic sensors, magnetic compass, and front-facing camera. All robots involved in a trial are initially placed near the beacon. Robots are programmed to stay within a 5 m ‘virtual fence’ of the beacon. In every experiment, 256 QR tags are arranged in 4 randomly placed clusters of 64 tags each.

Robot locations are continually transmitted over one-way WiFi communication to a central server and logged for later

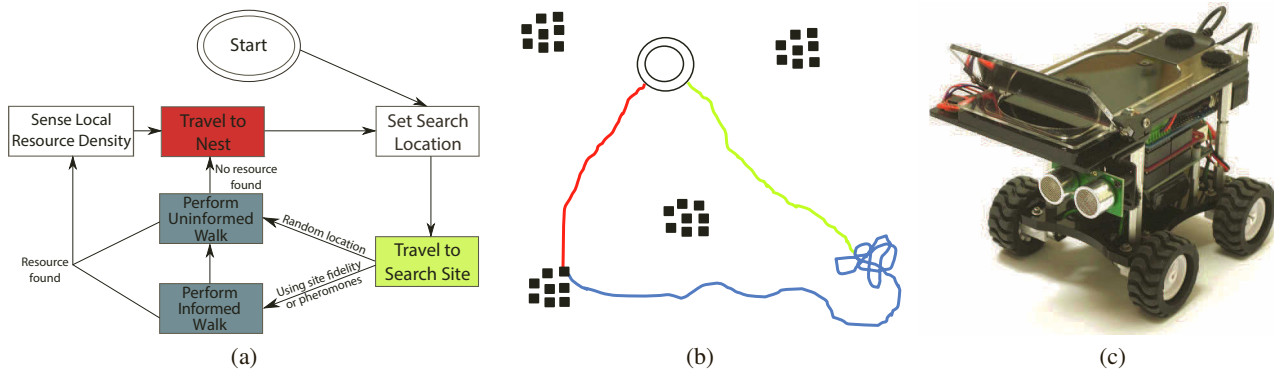


Figure 2: (a) State diagram describing the flow of behavior for individual robots during an experiment, (b) an example of a single cycle through this search behavior loop, and (c) an iAnt robot with Velcro for attaching reflective markers (motion capture was used for a previous experiment (Bezzo et al., 2013), but not for any of the observations in this paper). The robot begins its search at a central nest site (double circle) and **sets a search location**. The robot then **travels to the search site** (yellow line). Upon reaching the search location, the robot **searches for resources** (blue line) until a resource (black squares) is found. After sensing the local resource density, the robot **travels to the nest** (red line).

analysis. When a tag is found, its unique identification number is transmitted back to the server, providing us with a detailed record of tag discovery. Tags can only be read once, simulating seed retrieval. The central server also acts as a coordinator for pheromone waypoints using two-way communication. As each robot returns to the nest, the server selects a location from the list (if available) and transmits it to the robot.

Simulated teams of five robots search for resources on a 125 x 125 cellular grid. The system architecture replicates the physical dimensions of our real robots, their speed while traveling and searching, and the area over which they can detect resources. The spatial dimensions of the grid reflect the distribution of resources over a 100 m<sup>2</sup> physical area, and agents search for a simulated half hour. 256 identical resources are placed on the grid (each resource occupies a single grid cell) in one of three distributions: random (each resource placed at a random location), clustered (4 randomly placed clusters of 64 resources each), or power law (1 large cluster of 64, 4 medium clusters of 16, 16 small clusters of 4, and 64 randomly scattered). Each individual pile is placed at a new random, non-overlapping location for each fitness evaluation in an effort to avoid bias or convergence to a specific resource layout.

A population of 200 teams is evolved for 100 generations using recombination and mutation. Each team's parameter set is randomly initialized using uniform independent samples from each parameter's initialization function (see Table 1, column 3); agents within a team use identical parameters throughout the simulation. Each team forages for resources on its own grid, but the grids are identical. During each generation, all 200 teams undergo eight evaluations with different random placements of tag clusters; fitness is evaluated

as the sum total of resources collected by each team in the eight runs of a generation. Two individual teams are chosen through tournament selection and recombined through independent assortment: each parameter has a 10% chance of being selected from the second individual, otherwise it is selected from the first individual. Once selected, each parameter has a 10% chance of mutation.

We additionally conduct a series of *parameter swapping* experiments, in which we transfer a parameter set evolved in a simulated error-free world to a world with error. We compare the performance for parameters adapted to error to results using the original parameters not adapted to error. For these experiments, we average the resources collected across multiple replicates. In this way, we can determine the importance of including error in our model by testing whether it has a significant effect on the evolved behavior of the physical and simulated robot teams.

## Results

We present results for teams of five physical and simulated robots searching for resources in worlds with and without sensor error. Unless otherwise noted, results for each experimental treatment are averaged over five physical replicates and ten simulated replicates. Error bars indicate one standard deviation of the mean.

Figure 3 shows parameter values influencing robots' use of information ( $\lambda_{sf}$ ,  $\lambda_{lp}$ , and  $\lambda_{fp}$ ), as well as the informed walk decay rate ( $\lambda_{id}$ ) for random, clustered, and power law distributed resources. We observe similar values for all four parameters for clustered and power law distributions: robots evolve a high rate of following pheromones and a low rate of using site fidelity. Robots foraging on random distributions evolve both a high rate of following pheromones and

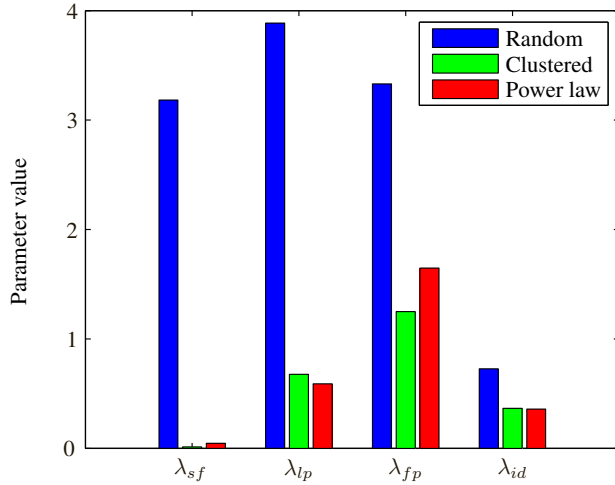
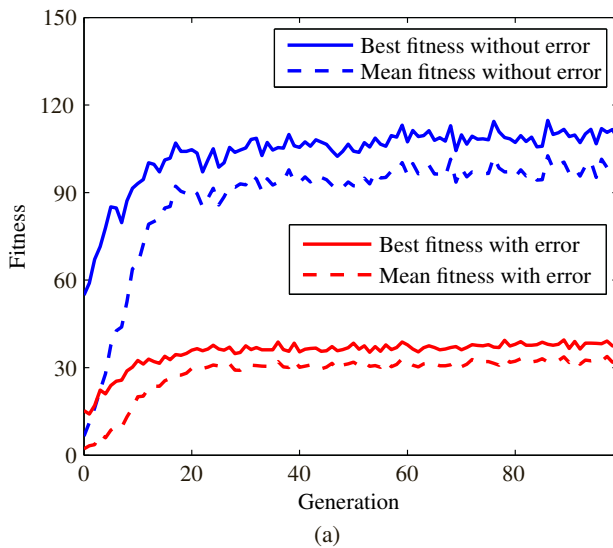


Figure 3: Parameter values for rates of site fidelity ( $\lambda_{sf}$ ), laying pheromone ( $\lambda_{lp}$ ), following pheromone ( $\lambda_{fp}$ ), and informed random walk decay ( $\lambda_{id}$ ) for random, clustered, and power law distributed resources.

a high rate of using site fidelity, but the effective probability of using either behavior are actually low because of the dependencies between them (see Algorithm 1).

Figure 4 shows fitness curves and parameter values adapted for simulated foraging for resources on a clustered distribution. Figure 4(a) plots best and mean fitness over 100 generations for worlds with and without positional and resource detection error modeled on our physical iAnt robots.



We observe fitness stabilizing after approximately 20 generations. Simulations with error converge to a fitness level approximately 33% of the fitness achieved in simulations without error. Figure 4(b) shows parameter values influencing robots' use of information ( $\lambda_{sf}$ ,  $\lambda_{lp}$ , and  $\lambda_{fp}$ ), as well as the informed walk decay rate ( $\lambda_{id}$ ). Robots foraging in an error-free world evolve a high rate of following pheromones (1.2) and a low rate of using site fidelity (0.013), whereas robots in a world with error evolve a high rate of site fidelity (1.7) and a low rate of following pheromones (0.0071). Additionally, in worlds with error, robots are 2.4 times more likely to lay pheromones, and their informed random walk decays 1.8 times faster than in an error-free world.

We analyze the performance of physical and simulated robots foraging in a world with error using parameters adapted specifically for the error-prone world. We compare the results to robots in a world with error using parameters adapted for an error-free world. Figure 5 shows the effects of parameter swapping on resource collection for physical and simulated robots (simulated results are averaged over 100 replicates). We observe an 80% improvement using the error-adapted parameters in physical robot teams, and a 16% improvement in simulated robot teams. We were able to distinguish a significant effect of parameter swapping in physical robots ( $t(8) = 5.1, p < 0.001$ ) and in simulation ( $t(198) = 17, p < 0.001$ ). Although simulated robots collect more resources than physical robots when using non-error-adapted parameters, we find that physical and simulated robots using error-adapted parameters are not significantly different ( $t(103) = 0.16, p = 0.87$ ).

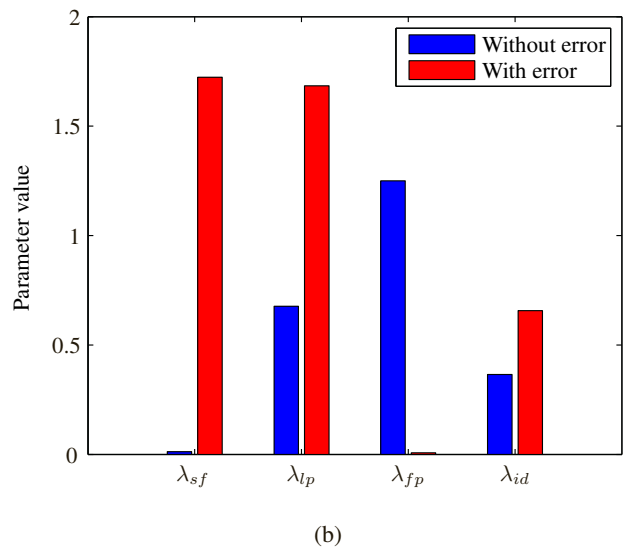


Figure 4: Results for simulated foraging on a clustered resource distribution with and without error. (a) Best and mean fitness curves. (b) Parameter values for rates of site fidelity ( $\lambda_{sf}$ ), laying pheromone ( $\lambda_{lp}$ ), following pheromone ( $\lambda_{fp}$ ), and informed random walk decay ( $\lambda_{id}$ ).



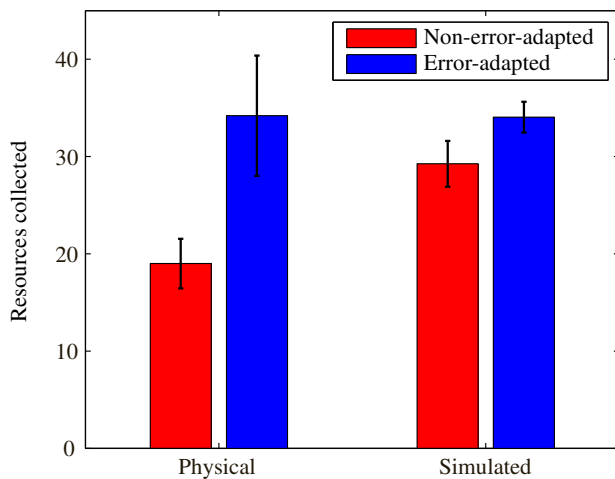


Figure 5: Results for physical and simulated robots foraging in a world with error using parameters adapted for a world with error, and parameters adapted for an error-free world. 80% more resources are collected using error-adapted parameters in physical robot teams, and 16% more are collected in simulated teams. Robots collected significantly more resources in both cases. Physical and simulated robots using error-adapted parameters are not significantly different.

## Discussion

Teams of physical and simulated robots used a central-place foraging algorithm (CPFA) to search for resources with and without sensor error. A genetic algorithm (GA) was used to evolve parameter sets which corresponded to robot team behaviors inspired by seed-harvester ants. We considered two types of error, positional error and resource detection error, and we explored the effects of error on overall resource collection and on individual evolved parameters. Error-adapted parameters improved performance of physical and simulated robots in worlds with error. We observed that teams of robots in error-adapted simulations collected resources at the same rate as physical robots.

Both positional and detection errors have the potential to confound a robot's ability to properly use information to exploit resources clustered via site fidelity or pheromones. Large positional errors in the estimation of resource locations can cause robots to perform informed random walks in regions without resources, thereby wasting time in detailed searches of the wrong areas. Errors in detecting resources can cause robots to underestimate the numbers of resources in a local area, so that robots fail to take advantage of memory or communication to return or recruit other agents to resource-rich locations.

Evolutionary algorithms have the potential to mitigate sensing errors by selecting for parameters which perform optimally given imperfect conditions. For example, robots

experiencing errors in resource detection benefit from a lower threshold of resource density detection for triggering creation of a pheromone waypoint. Robots with positional errors perform better with a faster decaying informed random walk, so that they quickly abandon detailed searches when there is a high probability that resources are not in remembered or communicated locations.

Parameter values for simulated robots foraging on random, clustered, and power law distributed resources (Fig. 3) illustrate the GA's ability to evolve sets of behaviors for each distribution. Parameters for clustered and power law distributions are similar, demonstrating the ability of the GA to focus on exploiting clumped resources when available. The lack of clustering in the random distribution induces the GA to effectively disable site fidelity and pheromone following behaviors, thus causing the adapted robot teams to concentrate on random exploration.

Fitness curves for simulations with and without error (Fig. 4(a)) demonstrate the ability of the GA to reliably converge. Parameter values (Fig. 4(b)) demonstrate the ability of the GA to evolve distinct sets of behaviors for an error-free world compared to a world with error.

Results for parameters swapped from error-free worlds into worlds with error (Fig. 5) show that parameters adapted for imperfect worlds outperformed parameters adapted for perfect worlds. Teams of physical and simulated robots collected similar numbers of resources, particularly when using parameters adapted for error. Thus, evolutionary methods effectively adapt robot behavior to sensor error. These results also mirror observations from our previous work in which genetic algorithms were used to evolve optimal parameter sets for specific types of resource distributions.

The work presented here motivates estimation of real robot error, evolution of parameters to fit with that error, and programming of those evolved parameters into real robots. In future work, we will conduct additional physical and simulated robot experiments using different numbers and distributions of resources, arena sizes, numbers of robots, and modes of communication to test whether simulations and physical experiments continue to correspond as closely as we have observed here.

## Acknowledgments

This work was funded by NSF EF #1038682 and DARPA CRASH #P-1070-113237.

## References

- Beverly, B. D., McLendon, H., Nacu, S., Holmes, S., and Gordon, D. (2009). How site fidelity leads to individual differences in the foraging activity of harvester ants. *Behavioral Ecology*, 20(3):633–638.
- Bezzo, N., Hecker, J., Stolleis, K., Moses, M., and Fierro,

- R. (2013). Exploiting Heterogeneous Robotic Systems in Cooperative Missions. *Robotica (in review)*.
- Brooks, R. A. (1991). New Approaches to Robotics. *Science*, 253(5025):1227–1232.
- Brooks, R. A. (1992). Artificial Life and Real Robots. In *Proceedings of ECAL*, pages 3–10.
- Curtis, S. A., Truszkowski, W., et al. (2003). ANTS for Human Exploration and Development of Space. In *IEEE Aerospace Conference*, volume 1, pages 1–161.
- Flanagan, T. P., Letendre, K., Burnside, W. R., Fricke, G. M., and Moses, M. E. (2012). Quantifying the Effect of Colony Size and Food Distribution on Harvester Ant Foraging. *PLoS ONE*, 7(7):e39427.
- Gage, D. W. (1995). Many-Robot MCM Search Systems. In *Autonomous Vehicles in Mine Countermeasures Symposium*, number April, pages 4–7.
- Hecker, J. P., Letendre, K., Stolleis, K., Washington, D., and Moses, M. E. (2012). Formica ex Machina: Ant Swarm Foraging From Physical to Virtual and Back Again. *Swarm Intelligence*, 7461:252–259.
- Hecker, J. P. and Moses, M. E. (2013). An Evolutionary Approach for Robust Adaptation of Robot Behavior to Sensor Error. In *GECCO 2013 (in press)*.
- Jakobi, N. (1997). Half-baked , Ad-hoc and Noisy : Minimal Simulations for Evolutionary Robotics. In *Proceedings of ECAL*, pages 348–357. MIT Press.
- Jakobi, N., Husbands, P., et al. (1995). Noise and The Reality Gap : The Use of Simulation in Evolutionary Robotics. *Advances in Artificial Intelligence*, 704-720.
- Kitano, H., et al. (1999). Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE SMC’99 Conference Proceedings*, volume 6, pages 739–743. IEEE.
- Kong, C. S., Peng, N. A., and Rekleitis, I. (2006). Distributed coverage with multi-robot system. In *Proceedings of ICRA*, number May, pages 2423–2429. Ieee.
- Letendre, K. and Moses, M. E. (2013). Synergy in Ant Foraging Strategies : Memory and Communication Alone and In Combination. In *GECCO 2013 (in press)*.
- Mallon, E. B. and Franks, N. R. (2000). Ants estimate area using Buffon’s needle. *Proceedings of the Royal Society B*, 267(1445):765–770.
- Matarić, M. J. (1994). Learning to Behave Socially. *Proceedings of SAB*, (617):453—462.
- Matarić, M. J. (1997a). Behaviour-based control : examples from navigation , learning , group behavior. *Journal of Experimental & Theoretical AI*, 9(2-3):323–336.
- Matarić, M. J. (1997b). Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4:73–83.
- Miglino, O., Lund, H. H., and Nolfi, S. (1995). Evolving Mobile Robots in Simulated and Real Environments. *Artificial Life*, 2(4):417–434.
- Moses, M. E. (2005). *Metabolic scaling from individuals to societies*. PhD thesis, University of New Mexico.
- Nelson, A., et al. (2004). Evolution of neural controllers for competitive game playing with teams of mobile robots. *Robotics and Autonomous Systems*, 46(135-150).
- Nolfi, S., Floreano, D., et al. (1993). How to evolve autonomous robots: Different approaches in evolutionary robotics. *Artificial Life*, 4:190–197.
- Panait, L. and Luke, S. (2004). Learning ant foraging behaviors. *Proceedings of ALIFE9*.
- Parker, L. E. (1998). ALLIANCE: An architecture for fault tolerant multirobot cooperation. *Robotics and Automation, IEEE Transactions on*, 14(2):220–240.
- Paz Flanagan, T., Letendre, K., Burnside, W., Fricke, G.M., and Moses, M.. How Ants Turn Information into Food. *IEEE Conference on ALife*, pages 178–185, 2011.
- Pinciroli, C., Trianni, et al. (2011). ARGoS : a Plugable, Multi-Physics Engine Simulator for Heterogeneous Swarm Robotics. In *IEEE/RSJ Proceedings of IROS*, number December 2010, pages 5027–5034.
- Pinciroli, C., Trianni, V., et al. (2012). ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm intelligence*, 6:271–295.
- Prabhakar, B., Dektar, K. N., et al. (2012). The Regulation of Ant Colony Foraging Activity without Spatial Information. *PLoS Computational Biology*, 8(8):e1002670.
- Singh, M. and Parhi, D. (2011). Path optimisation of a mobile robot using an artificial neural network controller. *Int’l Journal of Systems Science*, 42(1):107–120.
- Tunstel, E., Dolan, J., et al. (2008). Mobile Robotic Surveying Performance for Planetary Surface Site Characterization. In *Proceedings of PerMIS’08*, pages 200–205.
- Vaughan, R. (2008). Massively multi-robot simulation in stage. *Swarm Intelligence*.
- Watson, R. A., et al. (2002). Embodied Evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18.
- Wittlinger, M., et al. (2006). The Ant Odometer: Stepping on Stilts and Stumps. *Science*, 312(5782):1965–1967.

# Conditions for Outperformance of Recombination in Online Evolution of Swarm Robots

Christopher Schwarzer<sup>1</sup> and Nico K. Michiels<sup>1</sup>

<sup>1</sup>University of Tuebingen, Institute for Evolution and Ecology, Germany  
christopher.schwarzer@uni-tuebingen.de

## Abstract

Genetic recombination is commonly used in evolutionary algorithms and yet its benefits are an open question in evolutionary biology. We investigate when recombination is actually beneficial in the evolutionary adaptation of swarm robot behaviour in dynamic environments. In this scenario, artificial evolution has to deal with challenges that are similar to natural evolution: it must run online, distributed and evolve the genome structure. These requirements could diminish the benefit of recombination due to disruptive crossover. Using neural networks as robot controllers, we reduce this disruptiveness with an adaptive mate choice that evolves the probability of recombination and the genetic similarity of mates. In two experiments with a multi-agent simulation, we compare the adaptive performance of this approach with random recombination and pure mutation. Whereas both recombination treatments naturally outperform at low mutation rates, pure mutation achieves its best performance with high rates, where it also outperforms random recombination. The adaptive mate choice, however, achieves the same performance as pure mutation at high rates and outperforms when the network size is increased. We also found that treatments with recombination evolved smaller neural networks with fewer links.

## Introduction

It is a major challenge to give artificial, autonomous systems adaptive and problem solving capabilities. One approach to this problem attempts to mimic the impressive results of natural evolution by emulating its mechanisms like mutation, recombination and selection in so-called evolutionary algorithms (Eiben and Smith, 2003). The prevalence of recombination in evolutionary algorithms is interesting, given that it is not essential for an evolutionary process and because it is an ongoing discussion in evolutionary biology why sex and recombination is beneficial (Rice, 2002). Sex is considered costly because asexuals can potentially reproduce twice as fast (Maynard Smith, 1978). But sex is predominant in nature, a fact that numerous theories attempt to explain by attributing benefits to sexuality and recombination that compensate for those costs (West et al., 1999). One prominent argument is that sex accelerates adaptation to changing environments (Bell, 1982). In evolutionary algorithms however,

recombination is not considered costly because there is no actual reproduction and benefits have been shown in many cases (Eiben and Bäck, 1997; Doerr et al., 2008).

In this work, we investigate recombination in the case of evolution of swarm robot behaviour in dynamic environments. This case has special requirements for the evolutionary algorithm that make it more similar to natural evolution, and we wonder if recombination still exhibits clear benefits. In swarm robotics, many small robots are deployed with limited individual capabilities but the swarm can have emergent capabilities through cooperation (Sahin, 2005). Due to the difficulty of developing cooperative behaviour for swarm robots, evolution is often employed for this purpose (Haasdijk et al., 2010; Bredeche et al., 2010). One major challenge in swarm robotics is to make the swarm fully autonomous and adaptive so it can operate independently in the dynamic, real world — which is similar to what natural organisms do.

The special requirements of this challenge complicate the application of conventional evolutionary approaches, for example Evolution Strategies (Beyer and Schwefel, 2002). First, swarm robotics avoids using a central supervising instance because the robots should operate autonomously with only local information. Many evolutionary algorithms are centralized and not capable for such distributed operation. Second, to be able to deal with a-priori unknown and dynamic environments, constant adaptation is required which necessitates an online evolutionary algorithm (Agogino et al., 2000). Online evolution optimizes a system in parallel while it is deployed in its task. Every candidate evaluation is done in the local, variable conditions and affects total system performance whereas offline evolution can evaluate under repeated, constant conditions and deploy only the best solutions. And last, a large flexibility in evolvable behaviours is needed to deal with unknown environments. We use here neural networks, which are a well established approach for evolving robot behaviour (Floreano and Mondada, 1994). It has been shown that evolving the structure of the neural networks by adding and removing neurons and links increases the flexibility and is advantageous (Stanley and Miikkulainen, 2002). However, structural evolution

of neural networks complicates their recombination and can lead to disruptive macro mutations unless network structures are correctly matched, for example in NEAT (Stanley and Miikkulainen, 2002).

There exist evolutionary algorithms that address each of those requirements individually but not simultaneously. We have recently developed an approach to fill this gap for distributed online evolution with structural evolution of neural networks (Schwarzer et al., 2011). With this approach, we investigate here when recombination is beneficial in a simulation experiment with four robots in a foraging task. This is a very low number for a swarm, but it tests the effectiveness of the evolutionary algorithm even with limited resources. The approach can be scaled up and is applicable to real swarm robots because information is exchanged only locally. Our expectation is that recombination increases the speed of adaptation and that in a limited time, an unfit population increases its performance faster than when only mutation is used.

The mutation rate is a crucial parameter of this comparison because it is known that increasing the mutation rate increases the speed of adaptation until it reaches a point where performance is reduced, called the error threshold (Eigen, 1971; Ochoa and Harvey, 1999). We also include different neural network sizes in the comparison because it has been shown that longer genomes can reduce the error threshold (Ochoa, 2006).

## Material and Methods

The experimental setup is an extension of our previously published study (Schwarzer et al., 2011). Compared to the earlier work, the genome structure and evolutionary algorithm have been expanded to improve the performance of recombination.

### Genome and Neural Network

The genome encodes a recurrent neural network with a variable number of hidden neurons. Any connection between hidden neurons is possible. The employed neuron model uses the weighted sum of inputs with bias and sigmoid activation function.

In Figure 1, an overview of the genome structure is shown which differs from our earlier approach. The genome is now diploid with two homologous chromosomes. Each chromosome is an array of gene sites is used, each gene site can be free or occupied with a gene. This arrangement makes it possible to mutate the chromosome by relocating genes to free spots without shifting the absolute positions of other genes. In this way, crossover is simplified because the majority of genes in related chromosomes always line up at the same position. There are two different gene types: node gene and link gene. A node gene contains a node ID and a bias value for the activation function; it produces a node

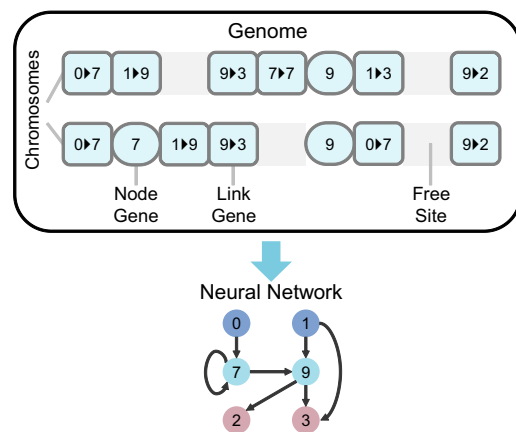


Figure 1: The genome has two chromosomes. A chromosome is an array of gene sites which can be occupied or empty. Genes come in two types, node genes and link genes. A node gene creates one node in the neural network, a link gene creates one link.

with the respective ID and parameters. A link gene contains two node IDs and a link weight; if there are node genes for both IDs, a neural link is created between them with the given weight. When new nodes are generated during mutation, they are assigned a random identifier. The identifier is used to recognize if two genes on the same gene site share a common ancestry.

A value can be calculated between two chromosomes that describes the genetic similarity  $s$  from 0 (completely different) to 1 (identical) in a similar way to the Sørensen Index (Sørensen, 1948). It is computed by comparing each homologous gene site, counting similar sites  $c$  and number of sites of both genomes ( $n_a$ ,  $n_b$ ) according to the following formula:

$$s = \frac{2 \cdot c}{n_a + n_b}$$

Genes are similar when they use the same node IDs. For the similarity between two genomes, the chromosomes are paired and the pairwise chromosome similarity values are averaged.

The genes of both chromosomes are used to create the neural network. Thus, it occurs regularly that there are multiple genes encoding the same structural element. This is normally the case for genes on homologous sites but also possible with genes across different sites due to relocation and recombination. These genes can have different parameters (weights, bias) for the same structure similar to different alleles in biology. In such cases, a dominance value is computed for the duplicate genes based on their parameter values. Dominant and recessive parameter values are evenly and finely distributed across the range of possible values. Only the allele with the highest dominance value is used and if multiple, different allele have the same dominance, their



parameters are averaged.

Outside of this chromosome structure, the genome contains three evolvable parameters. One mutation factor that is multiplied to the standard deviation of parameter mutation operations and to the probability of structural mutations. A mutation factor with a value of 1 leads to the default, unaltered mutation rate; values larger than 1 increase and smaller than 1 decrease it. The two other parameters are used for the adaptive mating mechanism explained in more detail in the description of the evolutionary algorithm.

Mutation of the genome is done on a per-gene-basis. Every gene is subjected to parameter mutation where weight and bias are mutated using a normal distribution with  $\sigma = 0.01$ . With a probability of 0.05, a gene undergoes structural mutation: a node gene changes its node ID to a new random ID and a link gene changes its source or destination ID to a different one that is already present on the genome. Furthermore, the gene may be completely removed ( $p=0.01$ ), duplicated with the copy being moved to a random position on the chromosome ( $p=0.01$ ) or relocated to a random, free site on the chromosome ( $p=0.02$ ).

An offspring genome is created similar to meiosis. First, a mutated copy of the whole genome is created for each parent. Then the chromosomes in each copy are crossed over with a probability of 0.02 per gene site. One resulting chromosome is picked from each parent and subsequently fused to form a diploid offspring genome.

## Evolutionary Algorithm

One instance of the evolutionary algorithm runs on each simulated robot and maintains an island population of ten genomes that serve as parental genome pool. In one cycle of online evolution, one offspring genome is generated from this genome pool and used to create a neural network. The neural network controls the robot for 2,500 simulation ticks (about 50 seconds in real time) during which it is evaluated based on the expressed behaviour. At the end of this time, the offspring genome may survive and replace one member of the island population, or it may be discarded, depending on its evaluation. Genomes also migrate between island populations independently of the evaluation cycle. When two robots are in close proximity of each other, one robot transmits and removes one random genome from its population; the recipient responds by sending a random genome back. This exchange occurs at most once every 10,000 ticks per robot. There are alternative approaches for this migration process which are possibly more effective, for example by only transmitting the best genome, but the given approach is sufficient for the purpose of this work in creating one virtual large population. The operation of this evolutionary algorithm is illustrated in Figure 2.

In order to create an offspring, one parent genome of the island population is selected randomly. We call this genome female here, but note that the genomes are equivalent to

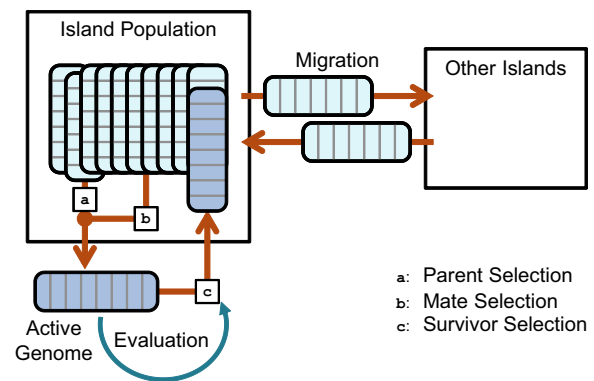


Figure 2: Operation of the distributed online evolutionary algorithm (Schwarzer et al., 2011). An island population is an instance of the distributed algorithm that runs on each robot. It maintains a constant number of genomes that serve as parental genome pool from which offspring genomes are created for evaluation. Genomes are occasionally exchanged between island populations.

hermaphrodites; each genome can have offspring with any other genome within the island population. The female can select a mate for recombination, depending on the employed mate selection strategy. If mate selection decides to not pick a mate, a mutated copy of the female is generated as offspring, which is always the outcome in the treatment without recombination. With random mating, mate selection picks a random genome from the island population (excluding itself). The adaptive mate selection strategy uses evolvable parameters of the female to influence the selection of the mate. First, the genetic similarity between the female and all other genomes of the local population is calculated. These similarity values are evaluated with a Gaussian function, with  $\mu$  and  $\sigma$  given by the female genome, to result in a value of “attractiveness” for each mate between 0 and 1. In other words, genomes have an evolvable value for the ideal genetic similarity of mates, and they can also evolve how much they are willing to deviate from this ideal. One mate is picked from the candidates with a random roulette selection, using the attractiveness as weights so candidates whose genetic similarity is closer to the ideal value have a larger likelihood of being chosen. However, it is also possible that no mate is picked when the sum of attractiveness of all candidates is less than 1. This can occur on purpose when the genome evolves, for example, a narrow range of acceptable values; it has then effectively reduced the probability of recombination.

In the survivor selection, which is identical in all three treatments, the freshly evaluated candidate genome is compared to the genomes in the island population and a special metric, which we call “fitness score” is used. The fitness score is the average of the last six offspring evaluations of a genome. A genome in the island population that did not

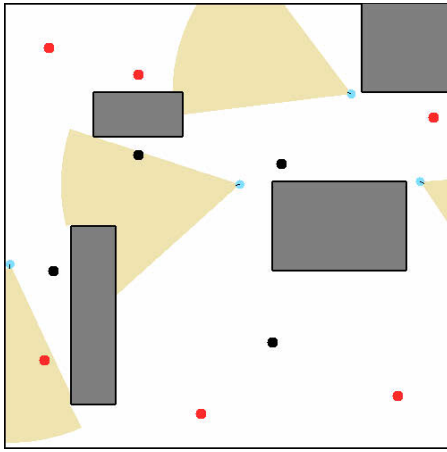


Figure 3: Screenshot of the experimental arena. The four simulated robots are shown with the approximate coverage of their visual sensors. Ten power stations are present that show their state by their colour: blinking red represents “charged”, black “depleted” or “charging”.

have six offspring yet, uses its own evaluation to seed this moving average. The candidate genome must have an evaluation greater than the worst fitness score in the population to survive and replace this worst genome. This mechanism makes the evolutionary algorithm tolerant to changes in the environment. A genome that once evaluated well will produce unfit offspring in a different environment, its fitness score drops as a result of low offspring evaluations and it can be more easily surpassed.

### Scenario and Simulation Environment

We use a foraging scenario that was designed to be practical and realizable in existing hardware. It requires search, identification and harvesting of power sources; an essential task of an autonomous swarm. The particular challenge for the neural network evolution is the processing of visual, colour information to distinguish resources from surroundings.

The experiment is done in the same 2D multi-agent simulation of our previous study (Schwarzer et al., 2011) and uses a similar scenario: four robots are in an arena with ten power stations that can be harvested by being next to them (see Fig. 3 for a visualisation of the arena). The stations charge only slowly and the best strategy is to continuously search the arena for nondepleted power stations.

The robots are modelled after a small, agile swarm robot like the e-puck (Mondada et al., 2009) or Wanda (Kettler et al., 2010) with a differential drive, distance sensors and a visual sensor that could be derived from an on-board camera. This visual sensor provides the robot with an RGB colour signal from three sectors, each  $20^\circ$  wide, covering together an area of  $60^\circ$  in front of the robot. One proximity sensors is also present in each of the three sectors.

All objects in the arena have a colour appearance. In order

| Factor                            | Values                 |
|-----------------------------------|------------------------|
| Recombination                     | None, Random, Adaptive |
| Initial Mutation Factor           | 0.031, 0.125, 1, 8, 32 |
| Adaptive Mutation                 | Disabled, Enabled      |
| Initial Neuron Count              | 0, 6, 14               |
| Colour Cycle Time (million ticks) | 0.1, 0.25, 0.5, 1, 2   |

Table 1: Factors of the experimental setup. Colour cycle time is only used in the colour cycle experiment.

to present a challenge that requires more complicated neural processing, the power stations are blinking by alternating their appearance between black and an active colour every five ticks. Depleted power stations do not blink and stay black until they have recharged some energy. A changing environment is simulated by changing the active colour of all power stations.

The current evaluation of a robot is increased every time it is next to an undepleted power station; at the same time, the current charge of that station is decreased. The time needed to fully drain a station is variable: the maximum time is 1,000 ticks, less time is needed when the robot decelerates. In this way, a robot can harvest from multiple stations within its 2,500 ticks of evaluation time, and the fastest way to gain reward is to come to a full halt next to a power station that is fully charged. Since a rarely visited power station is likely fully charged, exploration and exploitation of all power stations is promoted.

### Experimental Setup

The main experiment of this work is a two-stage scenario that we refer to as the change experiment: a randomly generated population is first evolved in one type of environment for a certain time, then the environment changes and the adaptation in the second stage is measured. Compared to using only a single stage, our preliminary runs have shown that this reduces the bias from choosing how the initial population is constructed and variance in the results. The two environments are created by changing the appearance of the power stations. In the first stage, the power stations blink in red, a unique colour signal in the arena and thus an easy challenge. After two million ticks, the colour changes to blue, which is the same colour as other robots and difficult to distinguish because the temporal change of the blinking signal has to be detected. We have shown with the same neuron model that the second stage requires recurrent connections and hidden neurons whereas top performance is possible in the first stage without hidden neurons (Schwarzer et al., 2011). The experiment continues for 4 million ticks in the second stage for a total of 6 million ticks.

In a second experiment, we test if different speeds of environmental change have an effect. The environment is

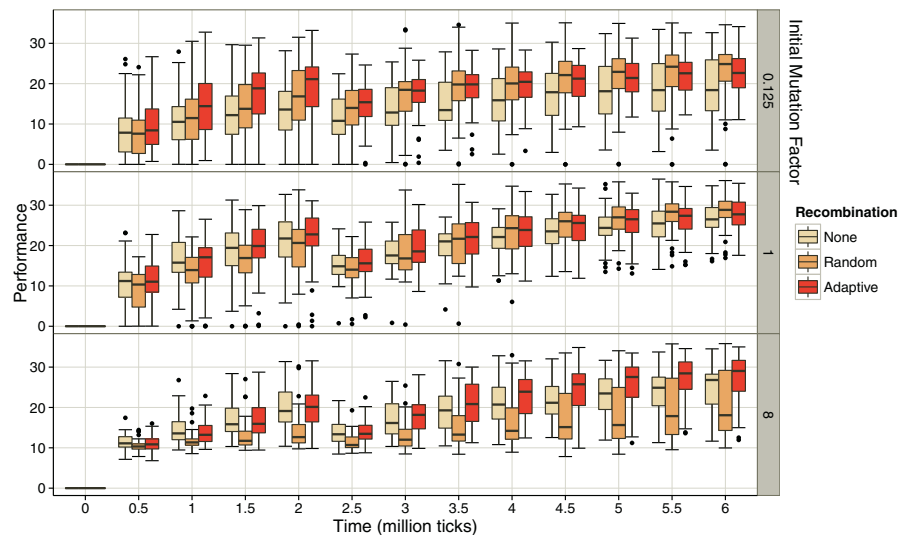


Figure 4: Development of collection performance of the change experiment. After the colour change at 2 million ticks, the performance drops sharply but the system re-adapts and performance recovers.

gradually changed by continuously altering the colour of the power stations by increasing the hue in HSV colour space at maximal saturation and brightness, starting with plain red. This leads to a continuous, repeating cycle through fully saturated colours. By changing the time for a full colour cycle, the speed of environmental change can be varied. The experiment lasts for 6 million ticks and we refer to this experiment as the colour cycle experiment.

The initial population for both experiments is generated with random genomes that encode a fully connected network with hidden neurons. All link weights are uniformly randomly initialized with values between  $\pm 0.2$ . One chromosome contains all necessary genes for these structures, it has twice as many gene sites as genes and they are placed on random sites. In addition to the genes for the connected network, six disconnected node genes are added so that treatments with zero initial neurons can still evolve them. In order to obtain a diploid genome, a mutated homologue copy of the first chromosome is added.

The mutation rate is controlled by the mutation factor on the genome. We let this mutation factor evolve slowly but also run trials with a fixed value. The number of initial neurons is varied, which affects the degree of structural evolution needed and the number of initial genes. Since the initial neurons are fully connected, the genome size increases quadratically with more neurons. The colour cycle experiment varies the time needed for a full colour cycle. See Table 1 for an overview of all experimental factors.

The response variable of both experiments, called (collection) performance, is the mean of all evaluations of all four robots in a time frame of 500,000 ticks. It is a fairly long

time, encompassing a total of 800 individual evaluations, in order to reduce variance. The theoretical maximum performance is 40, limited by the recharge rate of the power stations. Fifty replicates were done in all factor combinations.

## Results

Recombination naturally outperformed in the lower mutation factors of 0.031 and 0.125 but at these levels the absolute performance was also reduced. At higher mutation levels, the differences between pure mutation and recombination are small. The effect of the error threshold comes into play only with higher initial network sizes, and thus large genomes. With 0 initial neurons (32 initial genes per chromosome), performance is still high even with mutation factor of 32 whereas with 14 neurons (458 genes) some treatments have reduced performance at factors 8 and higher (shown in Figure 5). Evolving the mutation rate reduced this pattern only slightly. We use in the following result presentation the treatment with 14 initial neurons as a baseline because it shows the biggest differences. The adaptive mate choice generally settled at a low recombination rate of 14.7% ( $s = 16.8\%$ ).

The asterisk in the graphs indicate significance levels ( $p \leq 0.05$ : \*,  $p \leq 0.01$ : \*\*,  $p \leq 0.001$ : \*\*\*) of a two treatment comparison using Wilcoxon signed rank test with  $n = 50$ . Outliers in the box and whisker plots are more than 1.5 interquartile ranges outside of the box. The data was analysed using R Version 2.15.2 (R Development Core Team, 2011).

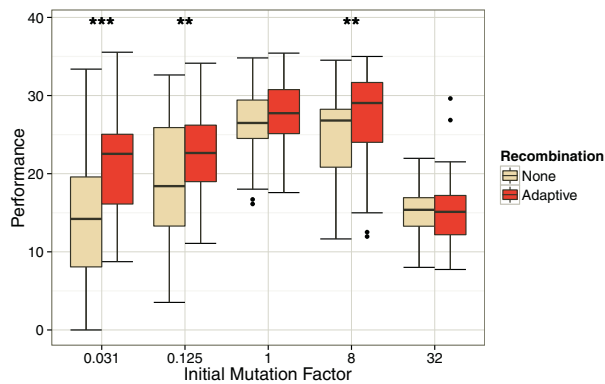


Figure 5: End performance of the change experiment with adaptive mate choice, adaptive mutation rate and 14 initial neurons. The effect of various mutation rates can be seen with an optimal rate between 1 and 8. In this configuration, recombination outperforms also at the elevated mutation factor of 8.

### Environmental Change Experiment

The two stages of the change experiment can be seen in Figure 4 where the development of performance over time is shown in the treatment with adaptive mutation rate and 14 initial neurons. After two million ticks, the performance drops sharply when the power stations switch their blinking colour from red to blue. The swarm re-adapts to the new situation and performance rises.

A comparison of collection performance at the end of the change experiment with 14 initial neurons, adaptive mate choice and adaptive mutation rate across mutation rates is shown in Figure 5. This is the only configuration where recombination significantly outperformed in mutation factors of 1 and higher ( $p \approx 0.114$  at factor 1,  $p \approx 0.004$  at factor 8).

This peculiarity is illustrated in Figure 6 where the relative performance of recombination versus pure mutation is shown across alternative configurations. (A) repeats the boxplots from Figure 5 at a mutation factor of 8, the other shown configurations keep this mutation factor and change one other experimental factor. In (B), random mating is shown instead of the adaptive mate choice. Random mating is actually performing better at low mutation rates, but at higher rates the results drop off to significantly lower levels than no recombination ( $p \approx 0.019$ ). The adaptive mutation rate is disabled in (C), the general performance drops and the significant difference between the two treatments disappears as well. In (D), 0 initial start neurons are used and end performance reaches peak levels but no significant difference are present. Thus, the advantage of recombination at high mutation levels is only seen with large networks, adaptive mutation rate and adaptive mate choice. However, the adaptive mate choice also never performed worse than without

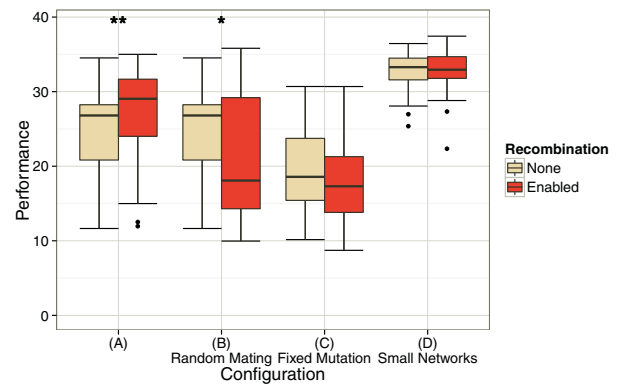


Figure 6: Comparison of the end performance of the change experiment with an elevated initial mutation factor of 8 in other configurations: (A) Same as Figure 5. (B) Random mating instead of adaptive mate choice. (C) Fixed mutation rate instead of adaptive rate. (D) 0 initial neurons instead of 14.

any recombination. Whereas random recombination suffers at high mutation rates and pure mutation suffers at low rates, the adaptive mate choice always reaches top performance.

We found an unexpected result when looking at the sizes of evolved neural networks, shown in Figure 7. Recombination treatments evolved generally smaller networks with fewer neural links. We find strongly significant differences for most of the parameter space we investigated, except in the most extreme mutation rates.

### Colour Cycle Experiment

The different rates of change in the colour cycle experiment have an inconclusive effect. End performance is similar across cycle times despite the large range of values covered. This can be seen in Figure 8 where the results with 14 initial neurons, initial mutation factor of 8, adaptive mutation rate and adaptive mate selection is shown. At the lowest colour cycle times of 100.000 ticks, the appearance of the power stations changes from one primary colour to the next within 14 evaluations but the system tolerates this rapid change well.

We find significant stronger performance of recombination at cycle time of 250.000 ( $p \approx 0.008$ ), 1.000.000 ( $p \approx 0.031$ ) and 2.000.000 ticks ( $p \approx 0.031$ ) but not at 100.000 and 500.000 ticks. As before in higher mutation rates, random recombination is significantly worse than the other two treatments. The resulting network sizes are also significantly lower here for recombination treatments, shown in Figure 9.

### Discussion

At the optimal mutation rate for the given scenario, pure mutation achieves similar performance as recombination. Random recombination virtually increases the effect of muta-



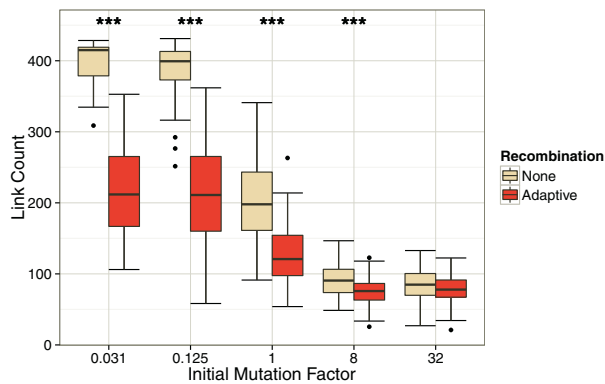


Figure 7: End link count of the neural networks in the change experiment. Recombination leads to smaller networks than no recombination except under extremely high mutation rates.

tion: compared to pure mutation, it has good performance at lower rates but suffers from increased mutation rates sooner. This agrees with the findings of Ochoa and Harvey (1999) where recombination reduced the error threshold. The reduced performance at higher mutation rates with larger initial networks is also similar to the results in Ochoa (2006) where the error threshold was lowered with increased genome length.

The adaptive mate choice mechanism, however, has always a strong performance and is robust to different mutation rates. It significantly outperforms in high mutation rates when the mutation rate itself can evolve and when using larger neural networks. This could be related to the effect of the error threshold as originally described in evolutionary biology: as genomes become larger, maximal mutation rate has to be reduced (Eigen, 1971). Our results indicate that recombination might be able to compensate for the reduced maximal mutation rate with selective mating strategies. And it is likely that better results are possible with more sophisticated mating selection.

Finally, the effect that recombination produces smaller networks is interesting since there are no direct costs for network size in the system. Also this seems unrelated to the effect of genome size as mentioned before because it is much stronger than the differences in performance. Also only the neural network is reduced; the number of genes stay roughly the same but the number of junk genes is increased. Large networks likely exacerbate the problem of network crossover, which is biggest for the random recombination treatment that also produces the smallest networks. Although the network size had no effect on performance here, smaller networks could easily be considered an advantage in terms of lower computational costs.

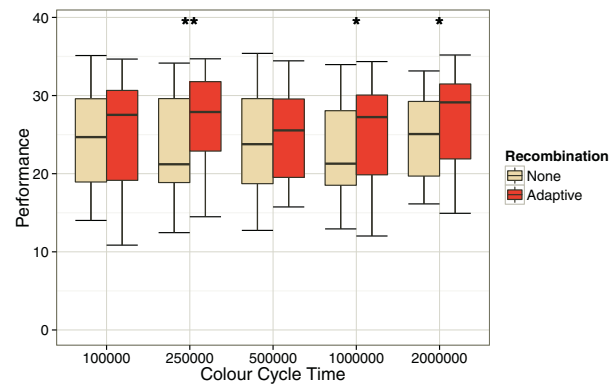


Figure 8: End performance of the colour cycle experiment. There is little effect of the different rates of colour change. Adaptive mate choice sometimes outperforms no recombination.

## Conclusion

In our simulated experiment about the online evolution of neural networks for swarm robots, we found that the difference in performance between pure mutation and recombination depends largely on the mutation rate and the size of the neural network. With the optimal mutation rate, omitting recombination achieved similar performance, but outside of this rate, recombination did outperform, in particular with large networks and adaptive mate selection. Since it can be difficult to estimate the optimal mutation rate before deploying a system, the robustness to suboptimal mutation rates can be an advantage.

Because recombination is not generally outperforming, our results contrast the conventional practice of generally including recombination in evolutionary algorithms and it might be worthwhile in some situations to omit it. The results are also an indication that the discussion in evolutionary biology about the benefits of sex is not completely unrelated to evolutionary computation. While recombination in evolutionary algorithms does not have the same costs of reproduction like sex in biology, it does suffer from genetic disruption due to breaking of favourable gene combinations, in particular with random recombination. We have shown that a mate selection strategy can mitigate this effect to improve recombination and outperform pure mutation even at elevated mutation rates. This advantage could become even bigger when our artificial organisms require larger, more complex genomes to handle more diverse environments.

## Acknowledgments

This work is part of project “SYMBRION” and is funded by the European Commission within the work programme “Future and Emergent Technologies Proactive” under the grant agreement no. 216342.

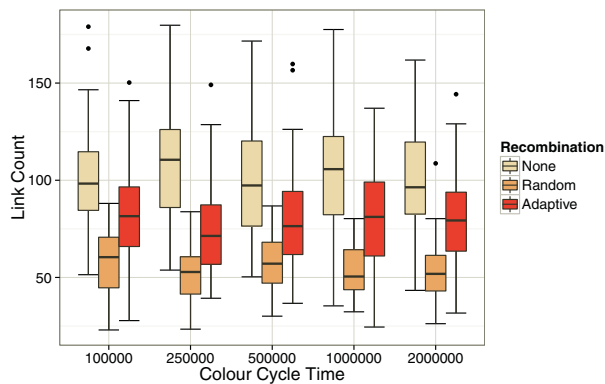


Figure 9: End link count of the colour cycle experiment. Random recombination produces the smallest networks, followed by adaptive mate choice and then no recombination.

## References

- Agogino, A., Stanley, K., and Miikkulainen, R. (2000). On-line interactive neuro-evolution. *Neural Processing Letters*, 11(1):29–38.
- Bell, G. (1982). *The Masterpiece of Nature: The Evolution and Genetics of Sexuality*. Croom Helm applied biology series. Croom Helm.
- Beyer, H. and Schwefel, H. (2002). Evolution strategies—a comprehensive introduction. *Natural computing*, 1(1):3–52.
- Bredeche, N., Haasdijk, E., and Eiben, A. (2010). On-line, on-board evolution of robot controllers. In *Artificial Evolution*, volume 5975 of *Lecture Notes in Computer Science*, pages 110–121. Springer.
- Doerr, B., Happ, E., and C., K. (2008). Crossover can provably be useful in evolutionary computation. *Conference on Genetic and evolutionary computation*, pages 539–546.
- Eiben, A. and Smith, J. (2003). *Introduction to Evolutionary Computation*. Natural Computing Series. Springer.
- Eiben, A. E. and Bäck, T. (1997). Empirical investigation of multi-parent recombination operators in evolution strategies. *Evolutionary Computation*, 5(3):347–365.
- Eigen, M. (1971). Selforganization of matter and the evolution of biological macromolecules. *Naturwissenschaften*, 58(10):465–523.
- Floreano, D. and Mondada, F. (1994). Automatic creation of an autonomous agent: genetic evolution of a neural-network driven robot. In *Conference on Simulation of adaptive behavior*, SAB94, pages 421–430, Cambridge, MA, USA. MIT Press.
- Haasdijk, E., Eiben, A., and Karafotias, G. (2010). On-line evolution of robot controllers by an encapsulated evolution strategy. In *Congress on Evolutionary Computation*, pages 1–7.
- Kettler, A., Szymanski, M., Liedke, J., and Wörn, H. (2010). Introducing wanda - a new robot for research, education, and arts. In *Conference on Intelligent Robots and Systems*, pages 4181–4186.
- Maynard Smith, J. (1978). *The Evolution of Sex*. Cambridge University Press.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotcz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Conference on Autonomous Robot Systems and Competitions*, pages 59–65.
- Ochoa, G. (2006). Error thresholds in genetic algorithms. *Evolutionary Computation*, 14(2):157–182.
- Ochoa, G. and Harvey, I. (1999). Recombination and error thresholds in finite populations. pages 245–264. Morgan Kaufmann.
- R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Rice, W. R. (2002). Experimental tests of the adaptive significance of sexual recombination. *Nature Reviews Genetics*, 3(4):241–251.
- Sahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. In Sahin, E. and Spears, W., editors, *Swarm Robotics*, volume 3342 of *Lecture Notes in Computer Science*, pages 10–20. Springer Berlin Heidelberg.
- Schwarzer, C., Schlachter, F., and Michiels, N. K. (2011). Online evolution in dynamic environments using neural networks in autonomous robots. *Advances in Intelligent Systems*, 4(3-4):288–298.
- Sørensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter*, 5:1–34.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural network through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.
- West, Lively, and Read (1999). A pluralist approach to sex and recombination. *Evolutionary Biology*, 12(6):1003–1012.

# Influence of a Social Gradient on a Swarm of Agents Controlled by the BEECLUST Algorithm

Daniela Kengyel<sup>1</sup>, Ronald Thenius<sup>1</sup>, Karl Crailsheim<sup>1</sup> and Thomas Schmickl<sup>1</sup>

<sup>1</sup>University of Graz, Institut of Zoology - Artificial Life Laboratory, Graz, Austria 8010  
daniela.kengyel@uni-graz.at

## Abstract

Agents controlled by a swarm algorithm interact with each other so that they have collective capabilities that a single agent does not have. The bio-inspired swarm-algorithm “BEECLUST” has the aim to aggregate a swarm at the global optimum even if there are several local optima (of the same type) present. But what about gradients produced of different stimulus types? In this paper, we present the concept of “social stimuli”. We investigate how robots controlled by the BEECLUST-algorithm react to a social stimulus which is created by placing immobilized robots in the environment. It shows that the robots controlled by the BEECLUST-algorithm are able to react on a social stimulus within an environment with a global and a local optimum.

## Background & Motivation

A swarm intelligent system is based only on a few simple rules which lead to a complex collective behaviour (Beni, 2005). Nature provides many swarm intelligent systems for us, where we can adopt algorithms. In earlier experiments such swarm intelligent behaviour of young honeybees was investigated. The preferred temperature of young honeybees is 36°C which is also the main temperature in the brood nest of a bee-hive (Grodzicki and Caputa, 2005; Heran, 1952). This temperature is very important for the development of young honeybees and can be interpreted as a natural mechanism to confine them to the brood nest.

In previous experiments it was found that a single young honeybee (*Apis mellifera*) is mostly not able to locate the area with its preferred temperature, whereas a group of young honeybees is able to find the right spot collectively (Kernbach et al., 2009; Schmickl et al., 2008). Figure 1(a) and 1(b) show example-results from such experiments. Similar pictures are shown in (Bodi et al., 2012). From this swarm-intelligent behaviour an algorithm was derived: the BEECLUST algorithm (Kernbach et al., 2009). This algorithm is used in autonomous swarmrobots to find the global optimum out of several local optima. Such optimum could be a light gradient or - in our case - a temperature gradient. What makes the BEECLUST algorithm special is that it has only a few requirements: The robot needs

| Not Needed                               | Requirements                                                 |
|------------------------------------------|--------------------------------------------------------------|
| no explicit communication                | sensors for distance-measurement                             |
| no permanent measurement of the gradient | one single non-directional sensor for measuring the gradient |
| no memory                                |                                                              |
| no ego-positioning                       |                                                              |
| no knowledge of the environment          |                                                              |
| no complex navigation (only random-walk) |                                                              |

Table 1: Requirements and non-requirements of the BEECLUST algorithm

infrared sensors for distance-measurement and a sensor for measuring the gradient (eg. light- or temperature-sensors). Thus, robots controlled by the BEECLUST algorithm do not need or use abilities that are essential for many other robot-algorithms (table 1).

As there are only a few requirements, the algorithm is used more often in robot swarms in the last years (Schmickl et al., 2008; Kernbach et al., 2009; Kengyel et al., 2011; Arvin et al., 2011). The algorithm is also analysed very well by theoretical models (Hamann et al., 2012; Schmickl and Hamann, 2011; Schmickl et al., 2009; Hereford, 2010).

This paper deals with another feature of the BEECLUST algorithm: “Can the decision-making be influenced by an additional kind of gradient?” An easy way to create a second kind of gradient is to place immobilized agents into the arena and thus creating something we define as “social gradient”. In the BEECLUST algorithm there is only a minimal social component modelled which is the discrimination of an obstacle and another agent. However, we assume that the system reacts on the social stimulus without changing the original algorithm and therefore the minimal social component plays an important role.

(Acerbi et al., 2007) showed that a social component can improve the success of algorithms based on genetic evolution or individual learning. An experimental setup was created artificially to analyze the social component in the BEECLUST-algorithm: Although the optima are typically not known a priori, immobilized agents were placed into the suboptimum allowing us to understand the effects of the social component of the algorithm. By investigating this behaviour our goal is to create new hypotheses for the swarm research of young honeybees and other social species. Please note, that these experiments aim for an improved understanding of the BEECLUST algorithm and not improvement of the efficiency.

A short description of the BEECLUST algorithm is provided here, for details see (Kernbach et al., 2009; Schmickl et al., 2008). The algorithm works as follows:

1. The agents move around randomly until the agent detects an object in the front.
2. If there is an object, the agent has to distinguish between a collision with an obstacle or another agent:
  - (a) If the agent collided with an obstacle, the agent turns around and moves randomly again.
  - (b) If this is an agent-to-agent-approach, both agents measure the temperature and calculate the waiting-time which is dependent on the temperature.
3. If the waiting-time is over, the agents move around randomly again.

In the experiments described in (Kengyel et al., 2011; Kernbach et al., 2009; Schmickl et al., 2008), the BEECLUST algorithm was derived and tested with a single kind of gradient: a light- or temperature-gradient. The agents, regardless of the type of agents, were always able to determine the global optimum. In (Schmickl et al., 2008) the algorithm is also tested with two gradients of different intensity and also in dynamically changing environments, showing that the algorithm is flexible enough to react on these dynamic changes of the environment. The cooperation of two swarms with different waiting time curves is investigated: It is shown that the two swarms benefit from the cooperation with each other if there is only a small swarm (Bodi et al., 2012).

In this work we present the results of experiments in which we investigate how a second, different kind of gradient influences the decision-making of the bio-inspired swarm-algorithm. This paper deals with the following questions:

1. Is the clustering behaviour of the BEECLUST algorithm sensitive to a social gradient?

2. Does a swarm have to trade off between two different gradients?
3. How many social agents are necessary to influence the aggregation behaviour?

## Material & Method

We implemented the BEECLUST algorithm in a simulation environment and designed experimental settings that allows us to answer the questions.

### Implementation of the Algorithm

For the simulation we used a free multiagent simulation-platform in which we modelled the bio-inspired swarm-algorithm simulating the individual honeybees as autonomous agents. In this work we use a simplified simulation of a robot swarm controlled by the BEECLUST algorithm.

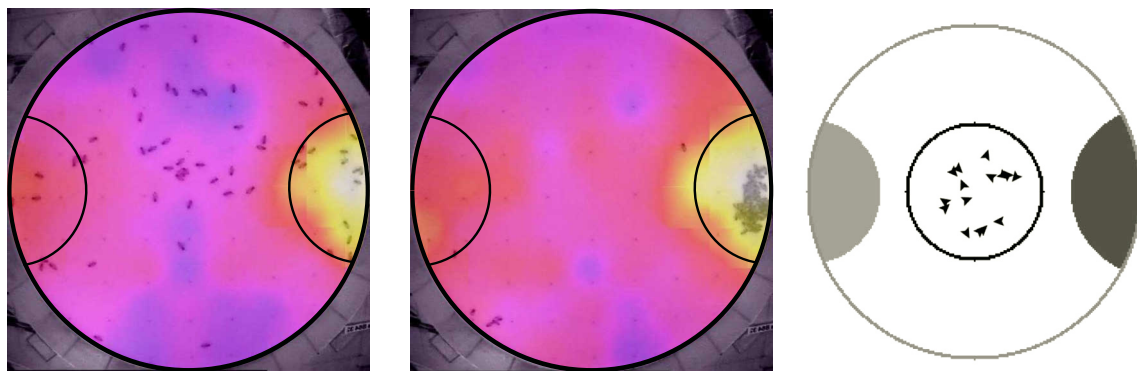
**Sensormodel** As the algorithm is derived from the behaviour of young honeybees we modelled their antennae for the measurement of temperature as follows: Each position in the arena has a specific temperature assigned (see section “Experimental Setup”). If an agent has to measure the temperature, it has to interrogate the temperature of its current position. This ensures an easy and efficient way to map a temperature and its measurement from an agent’s point of view. See section “Calculation of Waiting-time” for more detailed information.

As BEECLUST experiments were conducted with robots in (Schmickl et al., 2008) we modelled robots’ sensors for the distance measurement. Each simulated robot has three sensors: one in the front, one on the left and one on the right side. Each sensor has an aperture angle of 90°, so that the agent has a field-of-view of 270°. One robot has a diameter of 8 cm. To simulate the distance-measurement of robots we convert the measured distance into integer-values from 0 to 255. The visibility-range of an agent is about 1.5 robot-diameters. According to this a distance of 1.5 robot-diameters is mapped to a value of 0 and a distance of 0 robot-diameters results in a value of 255. A uniformly distributed random noise of [0;10] is added to the measurement.

| Temperature | Waiting-time |
|-------------|--------------|
| <26°C       | 0s - 10s     |
| 26°C - 29°C | 18s - 40s    |
| 29°C - 33°C | 18s - 80s    |
| >33°C       | 90s - 130s   |

Table 2: Dependency of local temperature and the waiting-time of the agent





(a) Bee-arena with bees at the beginning of the experiment. Left side: local optimum with 32°C, right side: global optimum with 36°C.

(b) Bee-arena with bees at the end of the experiment. Bees are clustering at gray area is the local optimum with the global optimum at 36°C on the right side.

(c) Setup of the arena. The left light-gray area is the local optimum with a temperature of 30°C - 32°C and on the right side the dark-gray area indicates the global optimum with a temperature of 30°C - 36°C. The black circle inside of the arena shows the area where the agents (black triangles) are released at the beginning of the experiment.

Figure 1: Experimental setup of bee experiments (left and middle) and similar experimental setup for robots (right).

**Calculation of Waiting-time** Choosing the right waiting time for specific temperatures is one of the most important parts of the BEECLUST algorithm. Kernbach et.al. (Kernbach et al., 2009) measured the waiting-time of bees after a bee-to-bee-approach dependent on the temperature of the bees' current position. Based on these measurements the waiting-time in the simulation is chosen as follows: For a temperature less than 26°C, the waiting-time is chosen randomly between 0s and 10s, for temperatures between 26°C and 29°C the agents' waiting-time is between 18s and 40s and for temperatures between 29°C and 33°C a waiting-time from 18s to 80s is chosen. For temperatures above 33°C the calculated waiting-time is between 90s and 130s (table 2).

**Experimental Setup** The experimental setup is based on an arena that was built for monitoring the behaviour of young honeybees in a complex temperature gradient (Szopek et al., 2008; Bodi et al., 2012). Figure 1(a) and 1(b) show the setup of the bee-arena. Figure 1(c) shows the robot-arena which has similar experimental setup. The arena has a diameter of 25 agent-lengths. It is surrounded by a wall in which the agents are able to perform their movements. On the left and on the right side there are two heat sources which create two different temperature-gradients (light- and dark-gray areas in figure 1(c)): The global optimum with a maximum of 36°C is located on the right side of the arena and occupies an area of about 11% of the arena. The temperature gradient ranges from the right side with 36°C to the middle of the arena with 22°C. The local optimum is located on the left side of the arena and occupies as well 11% of the arena. Here the maximum

temperature is 32°C on the left side and ranges to the middle of the arena with 22°C. The 30°C threshold is defined as the border of the local and the global optimum. 78% of the area (white area inside of the arena in figure 1(c)) is defined as the pessimum and has a temperature of 22°C in the middle of the arena.

To test the hypotheses we designed four different experiments (figure 2):

- (1) This experiment is used as a reference-experiment. Here we test the BEECLUST algorithm for the given arena with the global optimum on the right side and without the suboptimum on the left side (figure 2(a)).
- (2) In this experiment we additionally provide the suboptimum on the left side of the arena (figure 2(b)).
- (3) To test how a social stimulus affects the behaviour of aggregation in experiment 3 we place immobilized agents in the suboptimum with 32°C and a dummy-agent in the global optimum with 36°C to avoid side-effects (e.g. jamming-effects) (figure 2(c) and 2(d)). This experiment is conducted with different numbers of social agents to demonstrate how the system reacts to different sizes of a social gradient:
  - (a) with 1 agent acting as a social gradient (as depicted in figure 2(c)).
  - (b) with 2 agents
  - (c) with 3 agents and
  - (d) with 4 agents (as depicted in figure 2(d)).

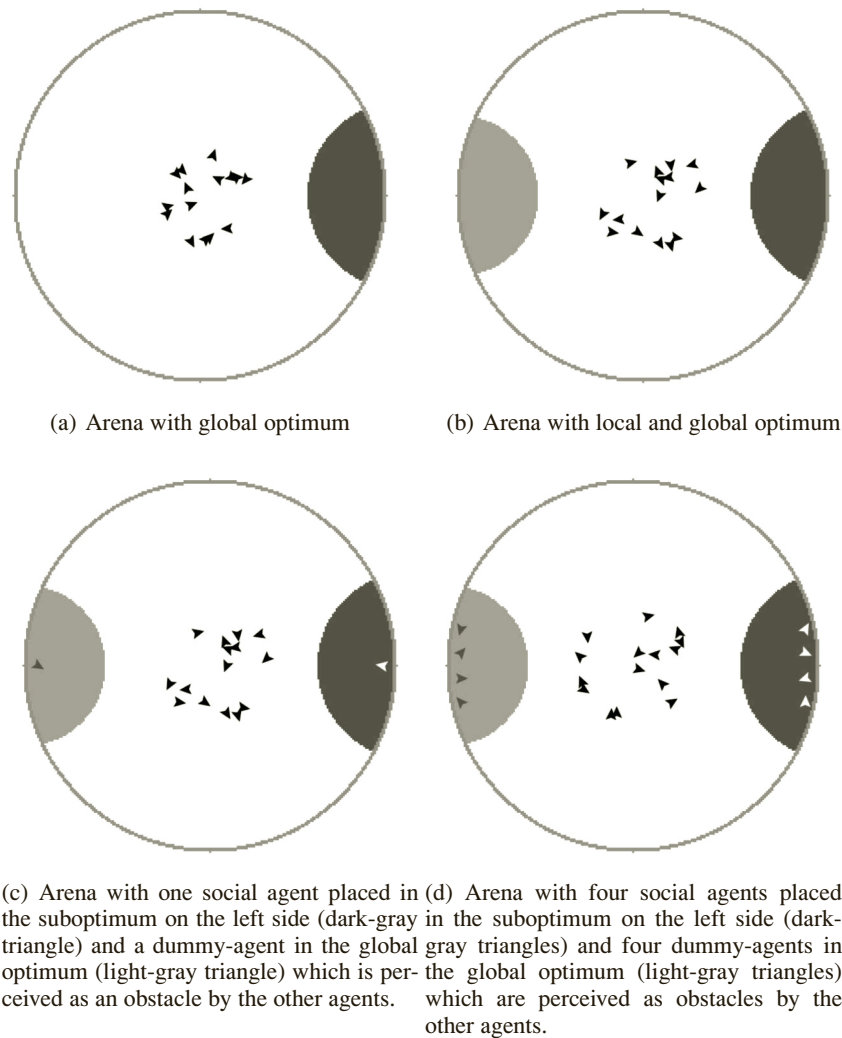


Figure 2: Different experimental settings. The dark-gray area on the right side of the arena indicates the global optimum with a temperature of  $30^{\circ}\text{C}$  -  $36^{\circ}\text{C}$ . The light-gray area on the left side is the local optimum with  $30^{\circ}\text{C}$  -  $32^{\circ}\text{C}$ . The black triangles indicate agents that are controlled by the BEECLUST algorithm.

Each experiment was repeated 100 times. At the beginning of each experiment the agents are placed randomly inside a central area which has a diameter of 10 agent-lengths and is located in the middle of the arena (figure 1(c)). In all six experiments 15 agents perform the BEECLUST algorithm with identical parameter settings. The agents move around in the arena with a speed of two agent-lengths per second. Agents which generate the social stimuli are immobile and do not perform the BEECLUST algorithm. To ensure that placing agents into the suboptimum has no side-effects (e.g. regarding jamming-effects due to overcrowding of an optimum) we also place dummy-agents into the global optimum which are perceived as obstacles and not as an agent.

**Changes to the original BEECLUST Algorithm as published in (Schmickl et al., 2008)** The BEECLUST algorithm is not changed in its sequence, only the input values for calculating the waiting time mentioned in table 2 where taken from a temperature gradient instead of a light gradient. We didn't have to adapt the algorithm so that it responds to the social gradient.

## Results

On the x-axes of figure 3 and 6 the abbreviation "LO" is referring to experiment (2) with only a global and a local optimum. "1SA", "2SA", "3SA" and "4SA" are referring to experiments with a social gradient with 1, 2, 3 or 4 social agents, respectively.

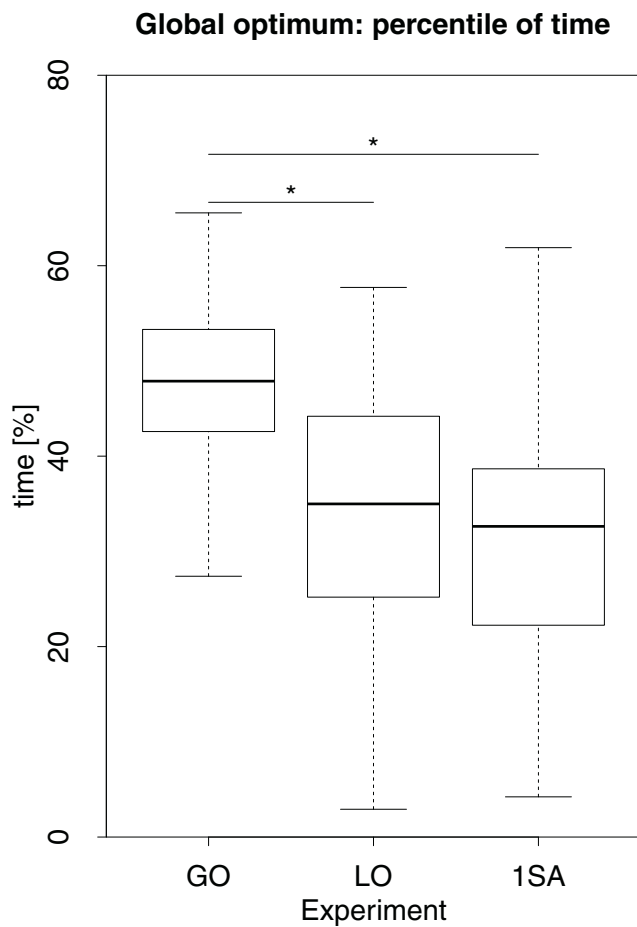


Figure 3: Percentile of time the agents spent in the global optimum. The plot shows the median with 1<sup>st</sup> and 3<sup>rd</sup> quartile.  $n=100$ . The bars with asterisks indicate significances at a significance-level of  $p=0.05$  and were tested with the Wilcoxon-Mann-Whitney-U test (nominal scaled).

The time the agents spent in the global optimum is shown in figure 3. In experiment 1 (“GO”) with just one optimum of 36°C the median time the agents spent in the global optimum is 47.88%. The median time in the experiment (2) with a local optimum (“LO”) is 34.98% and with 1 social agent (“1SA”) the median time is 32.62%. The results of experiment 1 is significantly different to the results of experiment 2 and 3a. The significances were tested with a level of  $p=0.05$  (Wilcoxon-Mann-Whitney-U test).

Figure 4 shows the time the agents spent in the global optimum of the experiments with a different amount of social agents. The median times of experiments with 1, 2, 3 and 4 social agents are 32.62%, 30.12% , 27.03% and 28.54%, respectively. Here, the significances were

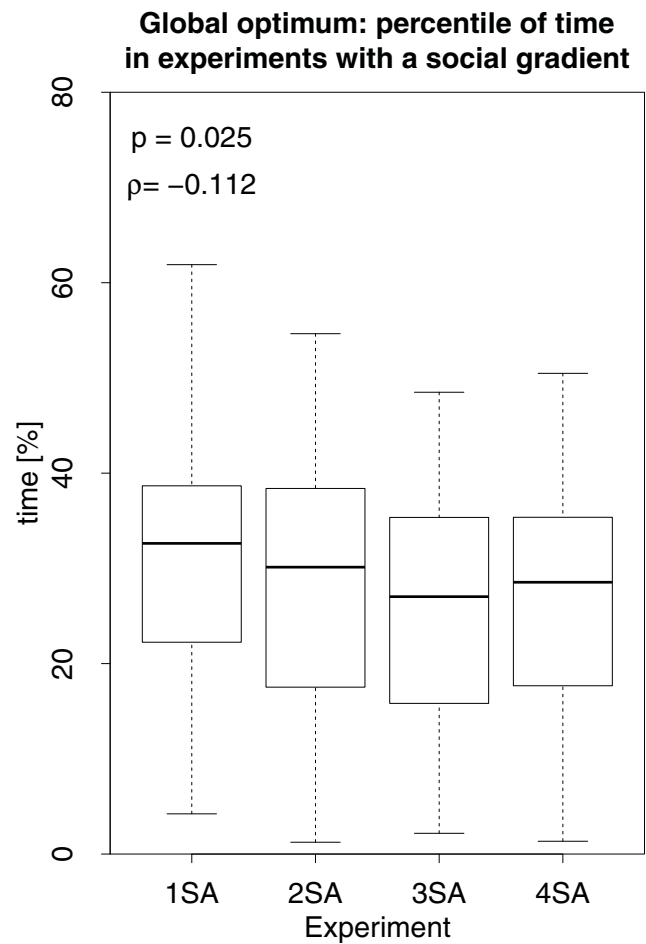


Figure 4: Percentile of time the agents spent in the global optimum in experiments with an increasing amount of social agents. The statistics is made with Spearman-statistics (ordinal scaled).

tested with Spearman-statistics and showed no significant correlation between the amount of social agents and the time the agents spent in the global optimum ( $p = 0.025$  and  $\rho = -0.112$ ).

Figure 6 shows the percentage of time the agents spent in the local optimum. The time for experiment 1 (“GO”) is calculated with a uniform distribution model, due to the fact that in this experimental setting no local optimum is available. As the defined area of the local optimum covers 11%, the agents would spend 11% of the time in this area. In experiment (2) the median time spent in the local optimum of 32°C was 16.25%. The median time for experiment (3a) with one social agent is 26.50%. The results of the experiments here are all significantly different to each other. The significances were tested with a level of  $p=0.05$  (Wilcoxon-Mann-Whitney-U test).

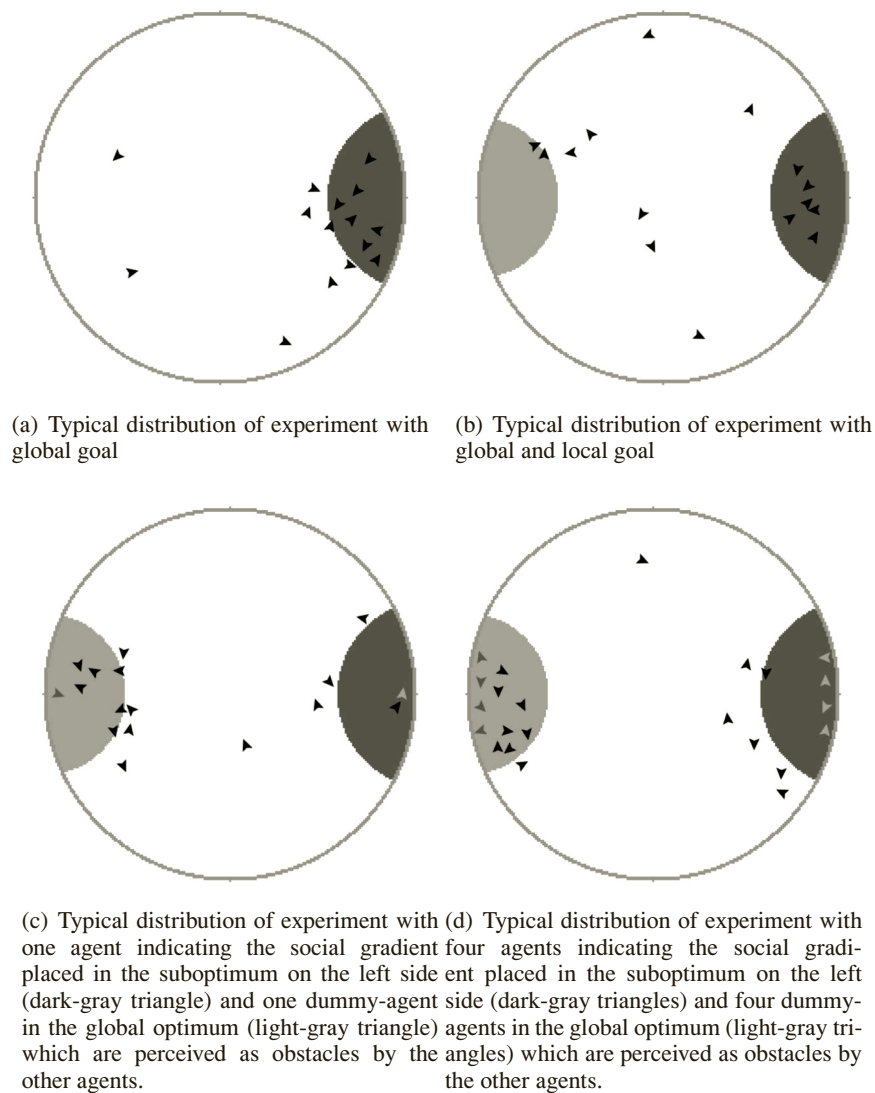


Figure 5: Examples of results of different experimental settings. The dark-gray area on the right side of the arena indicates the global optimum with a temperature of 30°C - 36°C. The light-gray area on the left side is the local optimum with 30°C - 32°C. The black triangles indicate agents that are controlled by the BEECLUST algorithm.

In figure 7 the time the agents spent in the local optimum are shown for the experiments with an increasing amount of social agents. The median times for 1, 2, 3 and 4 social agents are 26.50%, 27.08%, 29.21% and 30.04%, respectively. The significances were tested with Spearman-statistics and showed no significant correlation between the amount of social agents and the time the agents spent in the local optimum ( $p = 0.003$  and  $\rho = 0.148$ ).

## Discussion

The main feature of the BEECLUST algorithm is to find the global optimum within a complex environment, as shown

in (Schmickl et al., 2008) experiments with light-gradients in a dynamic environment were conducted. In the following, we will discuss the three questions mentioned in section “Background & Motivation”:

*Is the clustering behaviour of the BEECLUST algorithm sensitive to a social gradient?*

The BEECLUST algorithm as tested in (Schmickl et al., 2008) is able to locate the global optimum in static and dynamic environments robust. In the experiments we showed that this stable decision-making can be influenced by adding another gradient - a social gradient. By just using one additional agent - functioning as a source of a social



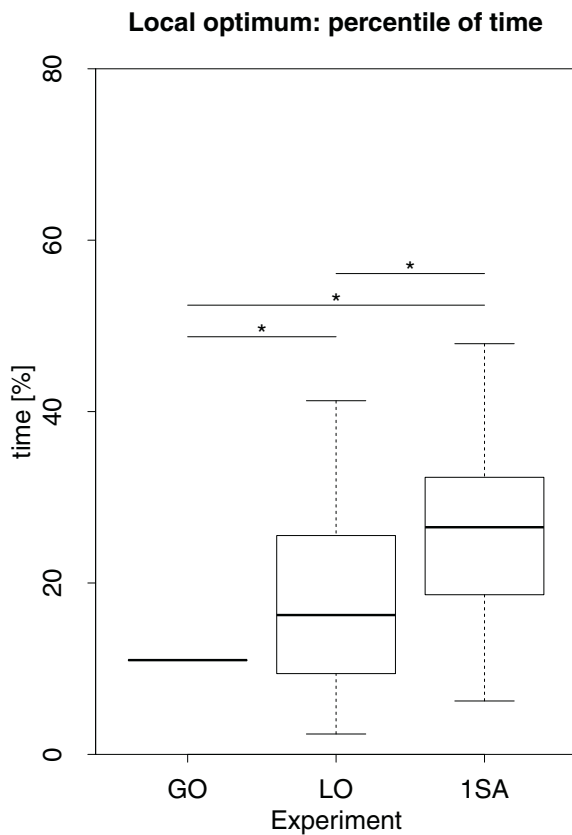


Figure 6: Percentile of time the agents spent in the local optimum. The plot shows the median with 1<sup>st</sup> and 3<sup>rd</sup> quartile.  $n=100$ . The bars with asterisks indicate significances at a significance-level of  $p=0.05$  and were tested with the Wilcoxon-Mann-Whitney-U test (nominal scaled).

gradient - we were able to increase the percentage of time in the local optimum from 16.25% to 26.50% (compare figures 5(b) and 5(c)).

*How many social agents are necessary to influence the aggregation behaviour?*

Adding just one single social agent had a huge effect. We were able to bound agents for more than 10% of the time to the suboptimum. To reach the threshold were the agents spend more time in the suboptimum than in the global optimum, three social agents were necessary (see figure 4 and 7). This leads us to the next question:

*Does a swarm have to trade off between two different gradients?*

A swarm of agents which is controlled by the BEECLUST algorithm always decides for the global optimum even if a second suboptimal gradient of the same type is present. Thus a discrimination of the local and the global optimum

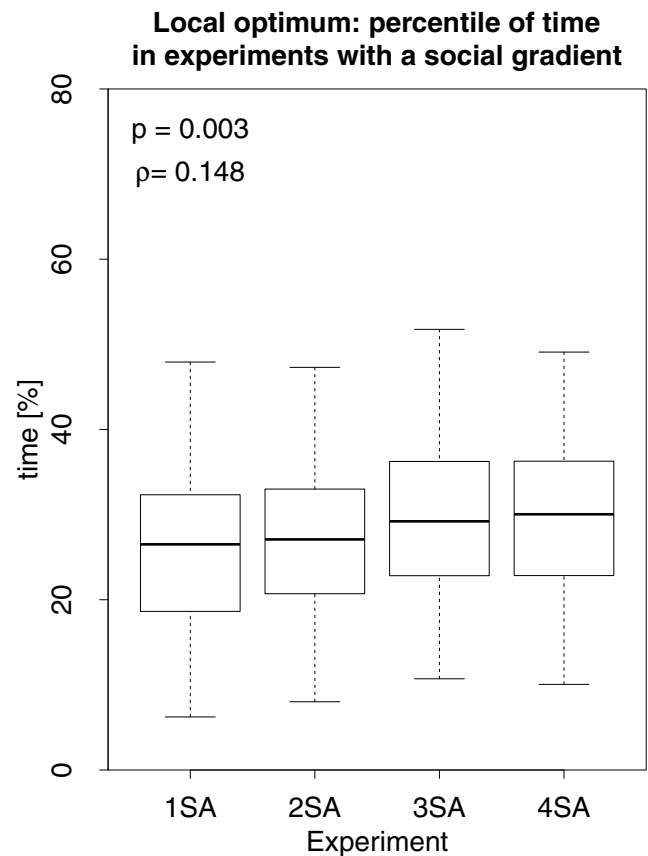


Figure 7: Percentile of time the agents spent in the global optimum in experiments with an increasing amount of social agents. The statistics is made with Spearman-statistics (ordinal scaled).

is possible. If there is a second gradient of another type, the decision-making of a swarm is not that clear anymore. In a weak gradient, agents which were undecided start to decide for the social gradient, but also agents from the global optimum reconsider their decision. If the social gradient gets stronger, no more agents are bound from the pessimum but some additional agents from the global optimum change their minds (see figure 5).

The percentage of time that agents spent in the pessimum is significantly lower in the experiments with social agents (3a, 3b, 3c) compared to the experiment without a social stimulus (2). Increasing the amount of social agents had no significant effects.

If we compare the results of experiment (3a) with experiment (2) it appears that fewer agents stay in the pessimum or global optimum and more agents stay in the local optimum. We can conclude that agents get bound not only from the global optimum but also from the pessimum (figure 3

and 6). This effect can also be observed in the results of experiment (3b). Three social agents is the minimum number of agents that are needed so that more agents place themselves in the local optimum than in the global optimum (figure 4 and 7). Adding another social agent - in total 4 social agents - leads to no significant changes in the percentage of time the robots spent in the optima.

## Conclusion & Future Work

We conclude that the BEECLUST algorithm can be influenced by using a social gradient induced by immobile agents placed in the suboptimal area of the arena.

As the social gradient had an unexpected big effect, we also want to introduce the social gradient to experiments with real honeybees. As the BEECLUST algorithm is derived from the behaviour of young honeybees, we expect that the decision-making of young honeybees can also be influenced by offering a second, different type of gradient. These results can then be used for further investigations of the swarm-intelligent behaviour of honeybees by creating bio-hybrid systems consisting of real honeybees and artificial autonomous robots (Schmickl et al., 2013).

In (Schmickl et al., 2008) robot-experiments were conducted in light-gradients. We want to implement the BEECLUST algorithm in robots and expose them to a temperature-gradient and a social gradient. With this, we want to get closer to the situation honeybees are faced with and provide feedback for the biological swarm research.

## Acknowledgments

This work is supported by the following grants: FWF (Austrian Science Fund) "REBODIMENT", no. P23943-N13; EU-ICT "CoCoRo", no. 270382; EU-ICT "ASSISL bf", no. 601074

## References

- Acerbi, A., Marocco, D., and Nolfi, S. (2007). Social facilitation on the development of foraging behaviors in a population of autonomous robots. In Almeida e Costa, F., Rocha, L., Costa, E., Harvey, I., and Coutinho, A., editors, *Advances in Artificial Life*, volume 4648 of *Lecture Notes in Computer Science*, pages 625–634. Springer Berlin Heidelberg.
- Arvin, F., Samsudin, K., Ramli, A. R., and Bekravi, M. (2011). Imitation of honeybee aggregation with collective behavior of swarm robots. *International Journal of Computational Intelligence Systems*, 4(4):739–748.
- Beni, G. (2005). From swarm intelligence to swarm robotics. In Şahin, E. and Spears, W. M., editors, *Swarm Robotics - SAB 2004 International Workshop*, volume 3342 of *LNCS*, pages 1–9, Santa Monica, CA. Springer-Verlag.
- Bodi, M., Thenius, R., Szopek, M., Schmickl, T., and Crailsheim, K. (2012). Interaction of robot swarms using the honeybee-inspired control algorithm beeclust. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):87–100.
- Grodzicki, P. and Caputa, M. (2005). Social versus individual behaviour: a comparative approach to thermal behaviour of the honeybee (*Apis mellifera* L.) and the american cockroach (periplaneta americana L.). *Journal of Insect Physiology*, 51(3):315 – 322.
- Hamann, H., Schmickl, T., Wörn, H., and Crailsheim, K. (2012). Analysis of emergent symmetry breaking in collective decision making. *Neural Computing and Applications*, 21(2):207–218.
- Heran, H. (1952). Untersuchungen über den Temperatursinn der Honigbiene (*Apis mellifica*) unter besonderer Berücksichtigung der Wahrnehmung strahlender Wärme. *Zeitschrift für vergleichende Physiologie*, 34:179–206.
- Hereford, J. M. (2010). Analysis of a new swarm search algorithm based on trophallaxis. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE.
- Kengyel, D., Schmickl, T., Hamann, H., Thenius, R., and Crailsheim, K. (2011). Embodiment of honeybee's thermotaxis in a mobile robot swarm. In *10th European Conference on Artificial Life (ECAL'09)*, volume 5777/5778 of *LNCS*. Springer-Verlag.
- Kernbach, S., Thenius, R., Kornienko, O., and Schmickl, T. (2009). Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic swarm. *Adaptive Behavior*, 17:237–259.
- Schmickl, T., Bogdan, S., Correia, L., Kernbach, S., Mondada, F., Bodi, M., Gribovskiy, A., Hahshold, S., Miklič, D., Szopek, M., Thenius, R., and Halloy, J. (2013). The path to assisi: Mixing social animals with robots to a bio-hybrid society. In *Living Machines - The International Conference on Biomimetic & Biohybrid Systems*, page in press. IEEE.
- Schmickl, T. and Hamann, H. (2011). Beeclust: A swarm algorithm derived from honeybees. *Bio-inspired Computing and Communication Networks. CRC Press (March 2011)*.
- Schmickl, T., Hamann, H., Wörn, H., and Crailsheim, K. (2009). Two different approaches to a macroscopic model of a bio-inspired robotic swarm. *Robotics and Autonomous Systems*, 57(9):913–921.
- Schmickl, T., Thenius, R., Möslinger, C., Radspieler, G., Kernbach, S., and Crailsheim, K. (2008). Get in touch: Cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18(1):133–155.
- Szopek, M., Radspieler, G., Schmickl, T., Thenius, R., and Crailsheim, K. (2008). Recording and tracking of locomotion and clustering behavior in young honeybees (*Apis mellifera*). In Spink, A., Ballintijn, M., Bogers, N., Grieco, F., Loijens, L., Noldus, L., Smit, G., and Zimmerman, P., editors, *Proceedings of Measuring Behavior 2008, Maastricht, August 26-29*, volume 6, page 327.

# Expanding the Time Horizon in Underwater Robot Swarms

Vincenzo Fioriti<sup>1</sup>, Stefano Chiesa<sup>1</sup>, Fabio Fratichini<sup>1</sup> and Andrea Arbore<sup>2</sup>

<sup>1</sup>ENEA, CR Casaccia, UTTEI-ROB Unit, Robotics Laboratory

<sup>2</sup>Aletea SpA, Rome, Italy  
vincenzo.fioriti@enea.it

## Abstract

In this paper we study the time delays affecting the diffusion of information in an underwater heterogeneous robot swarm in a time-sensitive environment. In many situations each member of the swarm must update its knowledge about the environment as soon as possible, thus every effort to expand the time horizon is useful. Otherwise critical information may not reach nodes far from the source causing dangerous misbehaviour of the swarm. We consider two extreme situations. In the first scenario we have an unique probabilistic delay distribution. In the second scenario, each agent is subject to a different truncated gaussian distribution, meaning local conditions are significantly different from link to link. We study how several swarm topologies react to the two scenarios and how to allocate the more efficient transmission resources in order to expand the time horizon. Results show that significant time savings under a gossip-like protocol are possible properly allocating the resources. Moreover, methods to determine the fastest swarm topologies and the most important nodes are suggested.

## Introduction

The robotic technology in ocean surveys, inspections, monitoring, pipe and cable tracking, has been well established in marine engineering (Leonard, 1998) with an important increase in performance in recent years (Nawaz, 2005).

Today, an AUV (Autonomous Underwater Vehicle) must be considered (Dell'Erba, 2012) as a real cost alternative to other available technologies, such as manned submersibles, remotely operated vehicles and towed instruments led by ships.

A group of underwater robots resembles closely a fish swarm, suggesting to use the properties of the biological swarm: coordinated movements, decentralized control, small interaction scale, minimal information broadcast.

The biologically inspired swarm control has advantages over the more complex but single robot: it covers a larger area, is fault tolerant, is self-aware. But it needs an inter-swarm information exchange and consequently delays in the information spread are generated (Beni, 1989).

However, communication channels are a major concern, as the acoustic underwater transmission is very slow and bandwidth limited, but, in the future, optical, high power transmission devices will be available for a number of different approaches integrating the acoustical data channel. Although optical methods are very powerful their

performances are affected by many strongly variable parameters like temperature, depth, salinity, turbidity, the presence of dissolved substances that change the colour and the transparency in different optical bands and the amount of solar radiation that heavily affect the signal to noise ratio. Moreover, horizontal underwater channels are prone to multipath propagation due to refraction, reflection, scattering.

The low speed of sound is also at the origin of significant Doppler effects, divided in frequency shifts and instantaneous frequency spreading contributing to the Doppler variance of received communication signals (Otnes, 2011).

Another important issue is the energy consumption, that requires optimization techniques to prevent batteries early exhaustion.

Nevertheless, the acoustic option for underwater data transmission is still the state-of-the-art methodology, and in this paper we will refer to it. To accomplish the above tasks it is necessary to control many submarine robots simultaneously, therefore researchers consider worthwhile to use biologically inspired models.

The swarm has advantages over the more complex but single robot usage: it covers a larger area, is fault tolerant. But it needs a heavy inter-swarm information exchange and consequently delays in the information spread.

## Goals

Depending on the local underwater environment, the speed of acoustic waves in water varies around 1500 m/s. The other source of delay is the management of the acoustic channel by proper MAC (medium access control) protocols (Otnes, 2011).

In this paper, we focus on the reduction of delays when the fast propagation of short warning messages to the whole swarm is needed. We consider the existence of a small group of AUV equipped with high quality communication devices able to reduce the transmission time with their closest neighbours.

Then the point is how to allocate these AUV into the suitable swarm configurations to obtain the largest time savings, extending the time horizon of the swarm.

To this end, we borrow from the graph theory some methods able to identify the most critical nodes with respect to the information transfer and test them numerically on several swarm configurations by means of Markov chains.

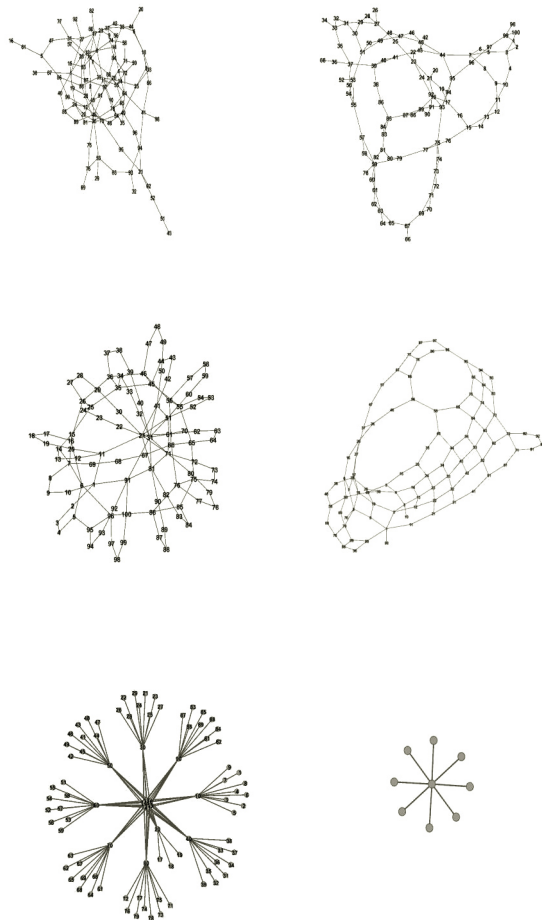


Figure 1: The swarm configurations. From top left: E-R, S-W, Cluster, Grid, Galaxy. Graphs have been modified with respect to the standard topologies. Bottom right: a single star.

Technical details of transmission underwater devices are out of scope, but can be found in a number of papers, for example see (Pompili, 2006), (Pompili, 2010) and (Burrowes, 2011).

## Scenarios and Data Preparation

### The Simulation Scenarios

Two scenarios are studied. In the first (scenario I), all delays are derived from an uniform probability distribution. In the second (scenario II), each link between two nodes is affected

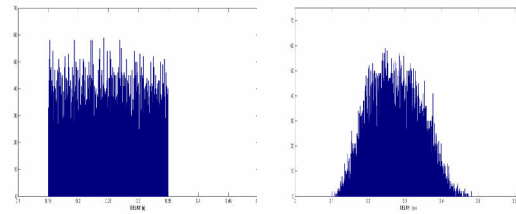


Figure 2: Left, delays uniform distribution (scenario I); right, delays positive support Gaussian distribution (scenario II).

from its own Gaussian distribution with positive support, because delays are strictly positive (Mazet, 2012).

A different probability distribution for each node pair  $i \rightarrow j$  has been implemented recently by (Picu, 2012), but not many researchers adopt this hypothesis, because of the analytical intractability of the mathematical related model.

Although we do not assume an explicit statistical dependence among links departing from the same node, we acknowledge the existence of a dependence (Chakrabarti, 2008) among processes that govern the delays (Xia, 2008). On the other hand, MAC protocols clearly indicate such dependences. This element is important because it rules out the usage of the Central Limit Theorem in the calculation of the overall graph delay and make it necessary the heuristic approach as opposed to the analytic approach.

The numerical values of the simulation have been elaborated according to a worst – case criterion from (Pompili, 2006); they cannot be considered typical of delays encountered in the underwater environment, as its variability and non stationarity are so wide to prevent any attempt, but can represent a reasonable case study. Also the Gaussian nature of the delay distribution is to be considered an approximation of reality, as statistics are sorely lacking.

The other major source of delay, the speed of the acoustic waves in water is taken into count in the simulation model, but in the scenario I the physical distances among nodes are fixed (150 m) while in scenario II may vary (150 – 225 m).

### The Configurations

The configurations assumed by the AUV (in the following called graphs or networks) considered in this paper are the random Erdos-Renyi (E-R), a graph whose nodes are connected almost randomly; the Grid, a regular disposition of nodes; the Small-World (S-W), a configuration between the grid and the random graph; the Galaxy, a group of interconnected star configurations; the Cluster, a group of interconnected ring configurations.

Note that in order to have the same number of links and node for all graphs, the basic structure of the graph has been modified adding/removing links. Hence, graphs of Figure 1 *do not resemble exactly* the standard topology of a small-world, a grid or a cluster.

Moreover, the links represent the acoustic communication channel between two nodes in a logical way, therefore the



geometrical form of graph in Figure 1 may change in the real environment, without loss of generality. These links are stochastic variables, meaning they may disappear randomly; their Euclidean length in the scenario I is set to 150 m, while in the scenario II can vary between 150 – 200 m, depending on a stochastic variable (if the link exists).

### The Optimization Methods

The number of the algorithms and of parameter devised to study graphs from the topological point of view is very large. Here we are interested in those who can determinate a small set of nodes (the “budget”) equipped with particular spreading influence. The phenomenon is analogous to the diffusion of a malware or a real virus. It is known (Arbore and Fioriti, 2013) that some nodes are more “important” and may facilitate or prevent the spreading. In most cases it is not a trivial task to find the influential nodes in large graph, but fortunately AUV swarms are reduced to less than a few hundreds. Nevertheless useful insight may be gained even with just one hundred of nodes, as in the present case.

The methods we use are: AV11, degree centrality, betweenness centrality and finally the random choice (in order to check results against the banal predictor). Others parameter are available, but these are probably the more relevant to our investigation. The degree centrality is the simplest algorithm. It suffices to count the number of links connected to a node. It is also intuitive that an high degree designate an “influent” node (called hub).

Betweenness centrality (BC): total number of shortest paths between every possible pair of nodes that pass through the given node. Betweenness looks for vertices connecting separated subgraphs (Newman, 2010), therefore high BC nodes may be understood as a sort of bridges.

AV11: selects a subset of  $k$  nodes all at once, according to spectral combinatorial methods. The selected subset may be optimal or suboptimal with respect to the brute-force method (Arbore, 2011). Spectral methods are able to analyze the dynamical behavior of graphs from the static appearance of the topology.

The algorithms identify ten nodes: for example, the degree centrality selects the upper row of nodes below and the AV11 the second row:

31, 21, 6, 61, 71, 81, 41, 91, 1, 51  
31, 21, 6, 61, 56, 81, 76, 36, 45, 15

in the case of the Cluster graph.

Once the set is identified, a high quality transmission AUV is assigned to the nodes of the graph. We simulate this operation assigning smaller delay capability to the links departing from the “optimized” node, i.e. optimally allocating the resources.

### The Protocol

The protocol we consider here is a gossip – like (sometimes called “epidemic spreading”) protocol, i.e. each node-agent in the swarm broadcasts the message to the one-hop neighbour.

The diffusion process is started by one node and must reach every node of the swarm within a finite time (the time horizon). In this paper we follow a number of specifications for the protocol, namely:

- the transmitting node contacts the receivers following the numeration priority rule (node  $j$  is contacted before node  $j+1$  (only single transmission are allowed);
- agents know only their neighbours;
- the receiving node has no previous knowledge of the message to be delivered;
- the link between two nodes  $i \rightarrow j$  is fixed, but the on/off status depends on a stochastic variable;
- if a link is off, no attempt to transmit is made;
- all particular delays due to MAC parameters are included into the stochastic delay characterizing the  $i \rightarrow j$  link;
- apart from re-transmitting the message eventually, no elaboration takes place inside the agent on the message data;
- the QoS (quality of service) of the communication link is completely described by the delay.

In practice, it is assumed the transmitting node contacts one neighbour at a time, sequentially.

This protocol is quite general to encompass many actual protocols. It is configured to serve as warning signalling tool, therefore the message to be transmitted is short (64B).

## Simulation Results

Numerical simulations have been conducted till stable results have been obtained. In Table 1 are shown the delays in the two scenarios. Absolute times should not be regarded as indicative of a general behaviour, because of the too wide variation of the conditions in the real marine environment. The important point here is the relative time difference among the topologies in each scenario. However, Table 1 shows that the variances are small and similar, hence the numerical experiments are reliable.

| Graph   | Average delay I | Delay Var. I | Average delay II | Delay Var. II |
|---------|-----------------|--------------|------------------|---------------|
| E - R   | 23.29           | 2.22         | 26.68            | 3.15          |
| S-World | 23.33           | 2.37         | 26.64            | 2.85          |
| Galaxy  | 23.32           | 2.16         | 25.70            | 3.00          |
| Grid    | 23.35           | 2.25         | 25.77            | 2.71          |
| Cluster | 23.34           | 1.87         | 25.75            | 2.86          |

Table 1: First two columns: overall delays and variances for different topologies with unique distribution scenario I (without optimization). Last two columns: overall delays and variances for different topologies (without optimization) in the multiple distributions scenario II. The time unit is the second.

In Table 2 the spectral analysis of the five topologies is presented. We rank them according to three well-known parameters of the spectral graph theory: the maximum eigenvalue of the adjacency matrix of the graph,  $\lambda_n$ , the spectral gap  $\lambda_{n-1} - \lambda_n$  of the laplacian matrix, and the algebraic connectivity  $\lambda_2$ , the first non zero eigenvalue of the laplacian matrix, given the ascending ordering:

$$\lambda_1 < \lambda_2 < \dots \lambda_{n-1} < \lambda_n$$

A large value of these parameters describe the connectivity of the graph, the lack or presence of isolated components, the lack or presence of bottlenecks (Restrepo, 2006) and (Van Mieghem, 2011). Information in well connected graph travel easily and the probability that a node may remain disconnected is lower, two important issues for AUV swarms.

The maximum eigenvalue of the adjacency matrix and the spectral gap are in complete accordance with the largest percentage reduction in the reduction of the overall delays (described in Table 3 and 4), while the algebraic connectivity differs only for the misclassification of the Grid and the Small-World topologies.

Therefore, the spectral analysis is also able to predict those swarm topologies prone to realize a major delay improvement allocating the best transmission resources in the nodes suggested by one of the topological algorithms.

Now, there is a trade-off on the largest eigenvalue of the adjacency matrix in the consensus problems of Olfati-Saber. According to (Olfati-Saber, 2004 and 2007) the stability of a fixed configuration is guaranteed iff:

$$\tau \leq \pi / 2\lambda_{max} \quad (1)$$

where  $\tau$  is the uniform delay experienced by the consensus distributed computations and  $\lambda_{max}$  is the maximum eigenvalue of the adjacency matrix.

Similar constraints may be set for non uniform switching topologies. The delay  $\tau$  depends on a number of factors: CPU power, data transmission bandwidth, MAC protocols, the number of AUV  $N$ , the inter-symbol interference. Clearly, it is convenient a small  $\lambda_{max}$ .

But, at the same time, we need to have a large  $\lambda_{max}$  to take advantage of the delay reduction in message transmissions. The spectral analysis of Table 2 therefore helps in determining the effect of  $\lambda_{max}$  on the delays.

| Graph   | Max Adj eigenval | Spectral Gap | Algebraic connectivity |
|---------|------------------|--------------|------------------------|
| Galaxy  | 7.9426           | 4.56030      | 0.16927                |
| Cluster | 5.1077           | 1.79950      | 0.12621                |
| E - R   | 3.9286           | 0.77056      | 0.10797                |
| Grid    | 3.4687           | 0.23352      | 0.01066                |
| S-World | 3.3473           | 0.15720      | 0.05807                |

Table 2: Spectral characteristics of the graphs examined. All graphs are of 100 nodes and 140 edges, 2.8 average degree. Simulations were at least of 1000 runs. All graphs are of 100 nodes and 140 edges. Topologies have been modified adding/removing edges to ensure the same number of edges and nodes and the same overall broadcasting delay within a  $\pm 0.2s$  interval. The length of the message to be transmitted is 256 bit in both scenarios.

Table 3 and 4 show the time saving gained using the four methodologies of optimization. Table 3 concerns the scenario I, where a unique uniform distribution provides all the delays between two communicating AUV (nodes of the graph). Only the relative percentage values should be considered, as the absolute values are strictly related to the particular framework examined. It is clear that the Galaxy (group of connected stars) configuration allows the largest delay reduction in both scenarios; furthermore, the star topology (Figure 1, bottom right) is well-known to guarantee good performance and reliability, although expensive.

The classification of the configurations according to the best delay after the optimization considering both scenarios is as follows: Galaxy, Cluster, Erdos-Renyi, Grid, Small-World.

It seems the Galaxy configuration of the AUV acoustic communications is the best by far. It is worth mentioning, anyway, that a star configuration relies heavily on the centre-

star to pass and sort messages, thus dissipating too much energy for the marine environment.

A solution is to alternate the centre-star role among the AUV of the star (Snels and Tabacchiera, 2013), possibly using a frequency coding to avoid signal overlapping, if enough high quality AUV are available. Otherwise the random Erdos-Renyi graph guarantees a good performance in a number of underwater tasks.

| Graph   | AV11                            | Degree                          | BC                | Random             |
|---------|---------------------------------|---------------------------------|-------------------|--------------------|
| E - R   | (15.90)<br>+0.78%               | <b>(15.76)</b><br><b>-32%</b>   | (16.83)<br>+6.7%  | (20.10)<br>+27.5%  |
| S-World | (18.89)<br>+9.3%                | <b>(17.28)</b><br><b>-25.9%</b> | (18.32)<br>+6%    | ((20.37)<br>+17.8% |
| Galaxy  | (9.28)<br>+143%                 | <b>(3.81)</b><br><b>-83%</b>    | (5.00)<br>+31%    | (18.88)<br>+395%   |
| Grid    | <b>(17.38)</b><br><b>-25.5%</b> | (18.75)<br>+7.8%                | (20.07)<br>+15%   | (19.52)<br>+9.16%  |
| Cluster | <b>(15.27)</b><br><b>-34.5%</b> | (16.38)<br>+7%                  | (16.21)<br>+6.15% | (19.90)<br>+30%    |

Table 3: Overall delay for different topologies in the single distribution scenario (I scenario), with optimization. Values within a  $\pm 0.2$  s confidence interval. The optimization techniques are: the AV11 algorithm, the degree centrality, the betweenness centrality and the complete random choice of nodes. Delays are expressed in seconds. Inside the brackets the absolute values. Percentage in bold is the best time saving

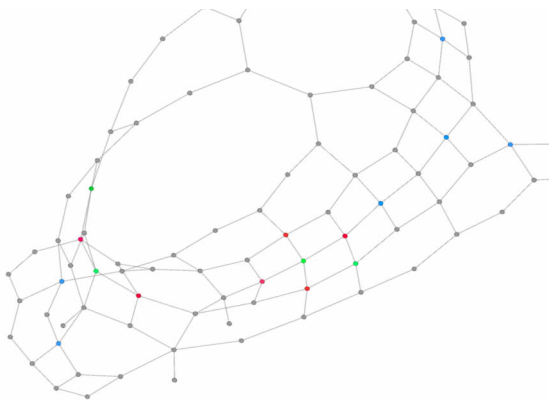


Figure 3: Comparison between degree centrality and AV11 for the case of the modified Grid graph. Green node have been selected by both algorithms, red nodes only by degree and blue

node only by AV11. An intuitive selection is not straightforward in this circumstance.

percentage with respect to the overall delays of Table 1 for a given graph.

Now we focus on the performance of the optimization algorithms (degree, AV11, BC and random choice) in selecting the most suitable nodes. Not surprisingly, in both scenarios the degree is the optimal choice in the case of E-R, S-W, Galaxy graph, but AV11 scores -7% with respect to degree for the Cluster and the Grid graph.

Then the simplest of the optimization algorithms (degree centrality) is not always the most successful, as an intuitive reasoning could expect. In fact, the AV11 exceed the degree in the Grid and in the Cluster cases while in the Erdos – Renyi case differences are really minimal.

When the configuration is as simple as the Galaxy (a group of stars) certainly the best choice is obviously the degree centrality parameter, but when the graph is more structured, more sophisticated algorithms such as AV11 are to be considered.

This is an useful finding, as often the static parameter (degree, betweenness, closeness etc.) are not able to identify correctly the influential nodes.

| Graph   | AV11                            | Degree                          | BC               | Random            |
|---------|---------------------------------|---------------------------------|------------------|-------------------|
| E-R     | (17.33)<br>+0.5%                | <b>(17.24)</b><br><b>-35.4%</b> | (18.54)<br>+7.5% | (22.19)<br>+28.7% |
| Small-W | (20.75)<br>+11%                 | <b>(18.68)</b><br><b>-29%</b>   | (20.12)<br>+7    | (22.39)<br>+19    |
| Galaxy  | (10.00)<br>+161%                | <b>(3.83)</b><br><b>-85%</b>    | (5.22)<br>+36%   | (20.98)<br>+447%  |
| Grid    | <b>(19.14)</b><br><b>-25%</b>   | (20.75)<br>+8.5%                | (22.44)<br>+17   | (21.16)<br>+11.5% |
| Cluster | <b>(17.56)</b><br><b>-31.8%</b> | (18.92)<br>+7.75%               | (18.51)<br>+5.4% | (22.36)<br>+27%   |

Table 4: Overall delays for different topologies with optimization in the multiple distributions scenario (II scenario). The optimization techniques are: the AV11 algorithm, the degree centrality, the betweenness centrality and the complete random choice of nodes. Delays are expressed in seconds. Inside the brackets the absolute values. Percentage in bold is the best time saving percentage with respect to the overall delays of Table 1 for a given graph.

## Theoretical Considerations

Chandra has shown how consensus and broadcasting are reducible to each other also in asynchronous networks with failure detectors (Chandra, 1996). Broadcast allows processes to distribute messages, so that they agree on the set of messages they deliver and the order of message deliveries.

The standard consensus form is:

$$x_i' = \sum_j a_{ij}(x_j - x_i) \quad , \quad j = 1, \dots, N \quad (2)$$

and its delayed version:

$$x_i' = \sum_j a_{ij}(x_j(t - \tau) - x_i) \quad , \quad j = 1, \dots, N \quad (3)$$

where  $a_{ij}$  are the entries of the adjacency matrix  $A$ .

It is known that for a connected network, the equilibrium point for (2) is globally exponentially stable. Moreover, the consensus value is equal to the average of the initial values; for small swarms the average is easy to calculate. In compact form (2) is written:

$$x' = -Lx \quad (4)$$

(2) and similar expressions are utilized in the swarm control to coordinate the states of the robots on a common position/velocity agreement resilient to disturbs (Olfati-Saber, 2007).

Since in the protocol section we have not required synchrony, now we can use the equivalence between the consensus problem and the broadcast problem (consisting in delivering a set of messages in the correct sequence to every agent of the network) to state a sufficient condition (Lu, 2011) for the successful broadcast of a set of subsequent short messages.

Lu shows that asynchronous consensus with stochastic delays can be obtained if during motion the swarm's graph has always had a spanning tree. The same result thus applies equally well to our broadcasting gossip-like protocol.

Recently (Atay, 2013, in press) a necessary and sufficient condition has been obtained for the discrete time, but only for fixed, non-switching links.

## Conclusions

We have examined the delays produced during the information diffusion in an underwater autonomous robot swarm in order to extend its time horizon, properly selecting the swarm configuration and the allocation of high quality transmission devices inside the configuration.

Since the independence condition of variables in the analytical treatment of the delay model cannot be assured generally, numerical simulations have been conducted.

Two scenarios have been simulated. Scenario I considers a unique uniform distribution of delays with fixed inter vehicle

distances; scenario II considers random distances and each link producing a delay, according to a positive support Gaussian distribution (as a delay is an inherently positive quantity). In both cases the links may or may not exist, depending on a uniform distribution probability.

Five swarm configurations (Galaxy, Cluster, Erdos-Renyi, Grid, Small-World) and four allocation methodologies (AV11, degree centrality, betweenness, random choice) have been tested to find the largest time savings.

Results show that the degree centrality applied to the galaxy configuration allows the largest delay reduction in both scenarios, as expected, but only when configuration are dominated by hubs. When the graphs are more structured, spectral allocation algorithms as AV11 are a better choice (Table 3 and 4).

Another useful finding is given by Table 2. The maximum adjacency eigenvalue and the spectral gap reveal the graph capabilities to decrease the delays when optimized.

**Acknowledgments.** This work has been supported by the HARNESS Project, funded by the Italian Institute of Technology (IIT).

## References

- Olfati-Saber, R., and Murray, R. M., (2004). Consensus Problem in Networks of Agents with Switching Topology and Time-Delays. *IEEE Trans. Aut. Contr.*, 49 (9) pp. 1520-1531.
- Olfati-Saber, R., et al., (2007). Consensus and Cooperation in Networked Multi-Agent Systems. *Proceedings of IEEE*, 95 1.
- Dell'Erba, R., and Moriconi, C., (2012). The Localization problem for Harness: a Multipurpose Robotic Swarm. *Sensorcomm 2012 Sixth International Conference on Sensor Technologies and Applications*.
- Pompili, D. and Melodia, T., (2005). Three-dimensional routing in under water acoustic sensor network. *PE-SWASUN '05 ACM Conference 1-59593, Montreal, Quebec, Canada*.
- Chiesa, S. and Taraglio, S., (2010). Flock Formation Control through Parameter Tuning in Underwater Swarms. *ICINCO'10 Conference Rome*.
- Snels, C. and Tabacchiera, M., (2013). Personal communication.
- Leonard, J. J., et al., (1998). Autonomous underwater vehicle navigation. *IEEE ICRA Workshop on Navigation of Outdoor Autonomous Vehicles*.
- Nawaz, S., et al., (2009). An Underwater Robotic Network for Monitoring Nuclear Waste Storage Pools. *1st International ICST Conference on Sensor Systems and Software*.



Arbore, A., Fioriti, V. A., (2013). Topological protection from the next generation malware: a survey. *International Journal of Critical Infrastructures*, (9)1:52-73.

Otnes, R., (2012). *Underwater networking techniques*. Springer Briefs in Computer Engineering, Springer.

Mazet, V., (2012). Simulation d'une distribution gaussienne tronquée sur un intervalle fini, Technical Report, Université de Strasbourg/CNRS.

Xia, C., et al, (2008). Epidemic spreading behaviour with time delay on local world evolving networks. *Front. Elect. Electron. Eng. China*, 3(2):129-135.

Lu, W. et al., (2011). Consensus and Synchronization in Discrete time networks of Multi-agents with Stochastically Switching topologies and Time-delays. *Networks and Heterogeneous Media*, (6)2:329-349.

Chandra, T., and Toueg, S., (1996). Unreliable failure detectors for reliable distributed systems, *Journal of ACM*, 43(2): 225-267.

Atay, F., (in press). The consensus problem in networks with transmission delays. To appear in: *Philosophical Transaction of the Royal Society A*.

Burrowes, G., and Khan, J., (2011). Short-Range Underwater Acoustic Communication Vehicles, Available from: [http://www.intechopen.com/books/autonomous\\_underwater](http://www.intechopen.com/books/autonomous_underwater)

Pompili, D., et al., (2006). Routing Algorithms for Delay-insensitive and Delay-sensitive Underwater Sensor Networks. *Mobi-Com, Los Angeles, California*, ACM 1 59593, 23:26.

Pompili, D., et al., (2010). Distributed Routing Algorithms for Underwater Acoustic Sensor Networks. *IEEE Trans. on Wireless Communications*, (9)9:2934-2944.

Newman, M.E.J. (2010). *Networks: An Introduction*. Oxford, UK: Oxford University Press.

Chakrabarti, D., (2008). Epidemic Thresholds in Real Networks, *ACM Transactions on Information and System Security*, (10)4:13.

Picu, A. Et al., (2012). An Analysis of the Information Spreading Delay in Heterogeneous Mobility DTNs, Technical report, ETH Zurich.

Restrepo J., et al., (2006). Characterizing the dynamical importance of network nodes and links', *Phy. Rev. Lett.*, 97:94-102.

Van Mieghem, et al. (2011) . Decreasing the Spectral Radius of a Graph by Link Removals, *Physical Review E*, 84: 016101.

Beni, G., and Wang, J., (1989). Swarm Intelligence in Cellular Robotic Systems, *Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy*.

# The Triangle of Life: Evolving Robots in Real-time and Real-space

A.E. Eiben<sup>1</sup>, N. Bredeche<sup>2,3</sup>, M. Hoogendoorn<sup>1</sup>, J. Stradner<sup>4</sup>, J. Timmis<sup>5</sup>, A.M. Tyrrell<sup>6</sup>, A. Winfield<sup>7</sup>

<sup>1</sup>VU University Amsterdam (a.e.eiben@vu.nl, m.hoogendoorn@vu.nl)

<sup>2</sup>UPMC Univ Paris 06, UMR 7222, ISIR, F-75005, Paris, France (nicolas.bredeche@upmc.fr)

<sup>3</sup>CNRS, UMR 7222, ISIR, F-75005, Paris, France

<sup>4</sup>Karl-Franzens University Graz (juergen.stradner@uni-graz.at)

<sup>5</sup>University of York (jon.timmis, andy.tyrrell@york.ac.uk)

<sup>6</sup>Bristol Robotics Lab, (alan.winfield@uwe.ac.uk)

## Abstract

Evolutionary robotics is heading towards fully embodied evolution in real-time and real-space. In this paper we introduce the Triangle of Life, a generic conceptual framework for such systems in which robots can actually reproduce. This framework can be instantiated with different hardware approaches and different reproduction mechanisms, but in all cases the system revolves around the conception of a new robot organism. The other components of the Triangle capture the principal stages of such a system; the Triangle as a whole serves as a guide for realizing this anticipated breakthrough and building systems where robot morphologies and controllers can evolve in real-time and real-space. After discussing this framework and the corresponding vision, we present a case study using the SYMBRION research project that realized some fragments of such a system in modular robot hardware.

## Introduction

Evolutionary robotics is heading towards fully embodied evolution in real-time and real-space. In this paper we introduce the Triangle of Life, a general conceptual framework that can help build systems where robots can actually reproduce. The framework can be instantiated with different hardware approaches and different reproduction mechanisms. For example, one could use classic mechatronic components and 3D-printing to produce new robots, or a stock of autonomous actuated robot modules as raw material and self-driven aggregation to implement ‘birth’.

The novelty of this framework lies in the pivotal role of reproduction and conception. The life cycle it captures does not run from birth to death, but from conception to conception and it is repeated *in real hardware* thus creating ‘robot children’ over and over again. This is new in evolved 3D printed robots, where the body structure is printed off-line. Even if the design is evolved, the printer only produces the end result after evolution is halted (in simulation), whereas in our framework printing=birth, thus being part of the evolutionary process, rather than following it.

Our approach is also new in self-assembling robot swarms, because existing work traditionally focusses on the transition of a swarm into an aggregated structure (a robot

organism) and vice versa. In the traditional setting, being aggregated is a transient state that enables the robots to meet a certain challenge after which they can disassemble and return to normal. In contrast, we perceive being aggregated as a permanent state and consider aggregated structures as viable robotic organisms with the *ability to reproduce*. That is, two or more organisms can recombine the (genetic) code that specifies their makeup and initiate the creation of a new robotic organism. This differs from earlier work aiming at *self-replication* and *self-reconfiguration* in that a ‘child organism’ is neither a replica of its parents, nor is it a reconfigured version of one of them.

This paper has a twofold objective, 1) to present the Triangle of Life as a conceptual framework for creating ALife of this type and 2) to illustrate how the components of this framework can be implemented in practice. To this end, we will use the SYMBRION research project<sup>1</sup> as a case study, even though originally the project only targeted traditional swarm-to-organism-to-swarm systems, cf. Levi and Kernbach (2010).

## Background and related work

The ideas in this paper can be considered from three perspectives, that of artificial life, evolutionary computing, and (evolutionary) robotics. The modern scientific vision of creating artificial life has a long history dating back to the 1987 Santa Fe workshop, cf. Langdon (1989); Levy (1992); Langton (1995). The most prominent streams in the development of the field are traditionally based on wetware (biology and/or chemistry), software (i.e., computer simulations), and hardware (that is, robots). In this paper we focus on the third option. The main contribution of the paper from this perspective is the introduction of a new, integrative framework, the Triangle of Life, that helps develop and study hardware-based ALife systems. In fact, the Triangle of Life defines a new category of ALife systems and outlines an interesting avenue for future research.

<sup>1</sup>EU Grant number FP7-ICT-2007.8.2, running between 2008-2013.

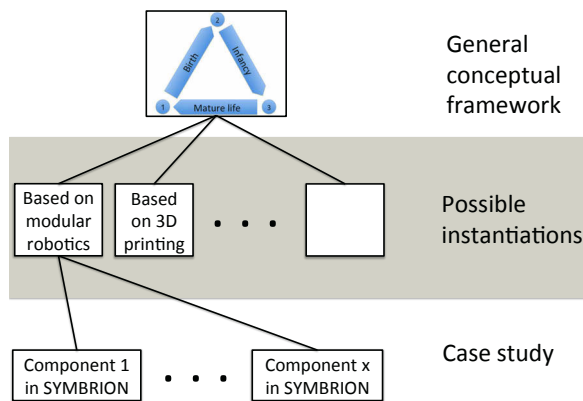


Figure 1: Positioning the Triangle of Life, its possible instantiations in general, and the specific examples used in this paper.

From an evolutionary perspective the framework we advocate here corresponds to a major transition from evolutionary computing (i.e., artificial evolution in software) to Embodied Artificial Evolution (i.e., artificial evolution in hardware) as introduced in Eiben et al. (2012). The roadmap outlined there considers embodiment in the broad sense, including biochemical approaches and treats mechatronics based embodied evolution as one of the possible incarnations. The work presented here represents the first detailed elaboration entirely devoted to that kind of systems.

Finally, the vision behind this paper can also be considered from the perspective of robotics. The relevant subarea here is evolutionary robotics that has a large body of related work, e.g., Nolfi and Floreano (2000); Wang et al. (2006); Trianni (2008). However, most existing systems in this field are based on simulations and use evolutionary algorithms as optimizers in an off-line fashion, during design time. Furthermore, evolution is usually applied to optimize/design some parts of the robot morphology or the controller, but rarely both of them. In contrast, our vision concerns real hardware, on-line evolution during run time, and it includes the evolution of both the morphologies and the controllers. In the system we envision, new robots are produced continuously only limited by the availability of the raw materials and the capacity of the ‘birth’ mechanism. In the resulting system evolution is not a simple optimizer of some robot features, but a force of continuous and pervasive adaptation.

In the landmark Golem project Lipson and Pollack (2000) evolved robots capable of moving themselves across a flat surface; robots were evolved in simulation and the fittest individuals then fabricated by first 3D printing the structural components then adding motors to actuate the robot. Although a remarkable achievement, the artificial creatures evolved then physically realized contained neither sens-

ing nor controller, so were not self-contained autonomous robots. Only the robot’s physical morphology was evolved.

The use of Lego has featured in evolutionary robot hardware. Although not evolving complete Lego robots work has been described, and indeed attempted to formalise the use of Lego structures for evolution. For example Funes and Pollack (1997) describe the simulated evolution, then construction using Lego, of physical bridge-like structures. Peysakhov et al. (2000) present a graph grammar for representing and evolving Lego assemblies, and Devert et al. (2006) describe BlindBuilder, an encoding scheme for evolving Lego-like structures.

Notably Lund (2003) describes the “Building Brains and Bodies approach” and demonstrates the co-evolution of a Lego robot body and its controller in which the evolved robot is physically constructed and tested. Here simulated evolution explores a robot body space with 3 different wheel types, 25 possible wheel positions and 11 sensor positions. Lund observes that although the body search space is small, with 825 possible solutions, the search space is actually much larger when taking into account the co-evolved controller parameters. This work is significant because it is, to the best of our knowledge, the only example to date of the simulated co-evolution, then physical realisation, of body morphology and controller for a complete autonomous mobile robot.

Work by Zykov et al. (2007) describes an evolving modular robotic system on the Molecube platform. In this work, self-reproduction is not a necessary prerequisite of evolution, but rather its target. In particular, the authors evolve self-replicators by employing a genetic algorithm (in a 2D simulation) where the measured amount of self-replication is used as an explicit fitness criterion to evaluate morphologies. Then, in a second stage they evolve a command sequence, i.e., controller, that enables a given morphology to produce an identical copy of itself. However, as yet, there is still no work that has fully demonstrated the online evolution of both structure and function of a modular robotic system, that is fully embodied in the modules themselves.

A related area with practical relevance to our vision is that of self-organizing robotic systems, Murata and Kurokawa (2012). Modular self-reconfigurable robot systems, cf. Yim et al. (2007), are particularly interesting because they constitute one of the possible technologies for implementing the Triangle of Life as shown in Figure 1. However, conceptually such systems are quite different from ours, because the emphasis is on self-reconfiguring morphologies to adapt to dynamic environments, whereas in our evolutionary system, new morphologies appear through ‘birth’ and adaptation of morphologies takes place over generations.

## The Triangle of Life

Throughout this paper we will not attempt to (re)define what life is. Instead, we take a pragmatic approach and con-

sider three features that are typically attributed to life or life-like systems: self-reproduction that relies on heredity, self-repair, and learning.

The proverbial Cycle of Life revolves around birth. We adopt this stance and define the Triangle of Life as shown in Figure 2.

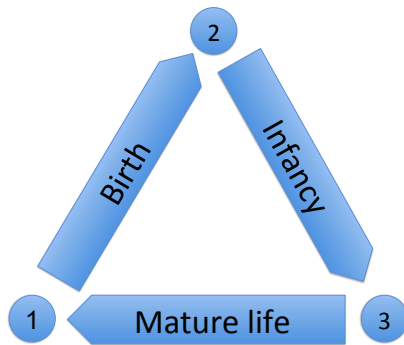


Figure 2: The Triangle of Life. The pivotal moments that span the triangle are: 1) Conception: A new genome is activated, construction of a new organism starts. 2) Delivery: Construction of the new organism is completed. 3) Fertility: The organism becomes ready to conceive offspring.

This concept of the Triangle is generic, the only significant assumption we maintain is the genotype-phenotype dichotomy. That is, we presume that the robotic organisms as observed ‘in the wild’ are the phenotypes encoded by their genotypes. In other words, any robotic organism can be seen as the expression of a piece of code that we call the genome. As part of this assumption we postulate that reproduction takes place at the genotypic level. This means that the evolutionary operators mutation and crossover are applied to the genotypes (to the code) and not to the phenotypes (to the robotic organisms). This fundamental assumption not only makes our envisioned systems more life-like, but –perhaps even more importantly– keeps the door open to enhancing the system with developmental abilities.

In the forthcoming subsections we will elaborate on each stage of the Triangle. For the sake of clarity we appeal to the modular robotic approach and explain some details in that setting. However, we emphasize that the Triangle is a generic framework equally applicable to modular and non-modular approaches.

### Birth

A new robotic organism is created first at genotype level and is thus seeded by a new piece of genetic code that is created by mutating or recombining existing pieces of code. Birth is therefore the first stage of life, specified as the interval between the moment of activating a newly created genome (circle 1 in Figure 2) and the moment when the robot organism encoded by this genome is completed (circle 2 in

Figure 2). In technical terms, this is the period when morphogenesis takes place. In principle, it can be implemented in various ways and later on we will illustrate some in detail. Here we suffice to distinguish two main categories, based on explicit vs. implicit representations of the shape of the newborn robot organism. Using an explicit representation, the genome explicitly specifies the shape of the organism and the process of morphogenesis is executed with this shape as target. Morphogenesis has therefore a clear stopping criterion; it is successfully completed when the target shape has been constructed. Using implicit representation the genome does not contain an exact description of the new shape. Rather, the genome can be seen as a set of rules governing the morphogenesis process that could follow different tracks and thus deliver different end shapes depending on the given circumstances and random effects. Note that this notion of implicit representation includes indirect, developmental representations, EvoDevo, ect. and connects our vision with the nascent area of morphogenetic engineering, cf. Doursat et al. (2012).

### Infancy

The second stage in the Triangle of Life starts when the morphogenesis of a new robot organism is completed (circle 2 in Figure 2) and ends when this organism acquires the skills necessary for living in the given world and becomes capable of conceiving offspring (circle 3 in Figure 2). This moment of becoming fertile is less easy to define in general than the other two nodes of the triangle. However, we believe it is useful to distinguish an Infancy period for two reasons. Firstly, the new organism needs some fine tuning. Even though its parents had well matching bodies and minds (i.e., shapes and controllers), recombination and mutation can shuffle the parental genotypes such that the resulting body and mind will not fit well. Not unlike a newborn calf the new organism needs to learn to control its own body. Depending on the given system design this could take place under protected circumstances, under parental supervision or within an artificial ‘nursery’ with a food rich environment, etc. From this perspective, the Infancy interval serves as a grace period that allows the new organism to reach its full potential. Secondly, the new organism needs to prove its viability. System resources are expensive, thus should be allocated to the creation of offspring with an expectedly high quality. Introducing a moment of becoming fertile (after birth) implies that organisms must reach a certain age before they can reproduce. From this perspective, the Infancy period serves as an initial assessment of implicit fitness that helps filter out inferior organisms before they start wasting resources by producing offspring.

The moment of becoming fertile can be specified by any user-defined criterion. This could be as simple as time elapsed after birth, or some measurable performance, for instance, speed (high is good) or amount of energy collected



(large is good) or number of collisions with obstacles (low is good), etc.

### Mature life

The third stage in the Triangle is the period of maturity. It starts when the organism in question becomes fertile (circle 3 in Figure 2) and leads to a new Triangle when this organism conceives a child, i.e., produces a new genome through recombination and/or mutation (circle 1).<sup>2</sup> It should be noted that at this point we switch perspectives: the beginning of a new life marks the beginning of another Triangle belonging to the new organism encoded by the new piece of genome. As for the ‘old’ organism nothing needs to end here. In other words, conceiving a child does not mean the end (death) of this organism, and it is certainly possible that an organism produces multiple offspring during its mature life. This view is motivated by the intuition behind the proverbial Cycle of Life that inspired our Triangle.

Robotic organisms can exhibit several behaviors during the mature period, depending on the given system and the interests of the experimenter. Here we will only consider two that we consider essential to any real world ALife system: reproduction and self-repair. Reproduction is an obvious requirement, but implementing it is challenging. For multi-cellular robotic organisms we see three feasible options:

1. Based on a ‘birth clinic’. After recombining the genomes of two parent organisms, the genome describing the new organism is beamed to a central facility where there are free robot modules. This is the place where the birth process is executed and a child robot is constructed.
2. Based on self-sacrifice. After recombining the genomes of two parent organisms, one of the parents disassembles and the child is built from its modules. Leftover modules become free riders and serve as available raw material. If the number of modules in the parent is not enough, others are recruited from such free riders.
3. Seed-based protocol. This will be discussed later as the one applied in SYMBRION.

Further to reproduction, we consider self-repair as an essential feature here. In simulation based ALife systems the world and its inhabitants can be stable and error-free, where randomness needs to be added deliberately. In the real-world systems we envision this is not the case, real hardware always breaks down. Thus, some form of self-repair is needed for continued operation after the inevitable breakdowns of the robot/organism. The ability to self-repair is linked to the

<sup>2</sup>Strictly speaking, the moment of producing a new genome need not be the same as activating this genome and starting the morphogenesis process, but this is just a formal detail with no real effect on the conceptual framework.

ability of the organism to perform morphogenesis, as it is very likely that some form of reconfiguration is needed in the event of failure.

### Implementing the Triangle of Life

As mentioned in the Introduction, originally the SYMBRION project considered robotic organisms as transient states of the system. An aggregated organism could achieve goals a simple swarm could not (negotiating an obstacle or reaching a power point) and after completion it could dis-aggregate again. However after five years of research and development many of the components that make up the Triangle of Life have been implemented in hardware or are very close to being implemented in the short term. The purpose of this section is to illustrate these achievements together and to indicate the current state of the art towards an integrated ALife system based on the modular robotic organisms concept.

#### Birth: Explicit Encoding for Morphogenesis

Within the Symbion framework a heterogeneous group of mobile robots can operate in *swarm* mode to – for instance – autonomously explore a region, exploiting the spatial distribution of the swarm. However, when required, Symbion robots can self-assemble to form a 3D *organism*. The process of transition from swarm-mode to organism-mode, with an explicit pre-defined (or pre-evolved) body plan, is also self-organising and proceeds as follows. Any individual robot in swarm mode can act as a ‘seed’ robot, initiating morphogenesis. Typically this might be when that robot discovers an environmental feature or hazard that cannot be accessed or overcome by individual swarm-mode robots. Each robot is pre-loaded with a set of body-plans, and the seed robot will select the most appropriate body plan for the current situation. The position of the seed robot in the selected body plan then determines the next robot(s) that need to be recruited by the seed robot, and the face(s) that they will need to dock into. The seed robot then broadcasts message bearing recruitment signals from the selected face(s), using the IR signalling system built into each docking face. That message specifies which of the three Symbion robot types needs to be recruited.

The autonomous docking approach is illustrated in Figure 3. Initially, a seed robot initiates recruitment of other robots. The pre-evolved body plan is then transferred from the seed robot to them, so newly recruited robots then determine their own position in the growing organism. In discovering its position a robot also determines whether or not other robot(s) need to be recruited. In Figure 3 image 2 they do. Robots’ recruitment signals can be detected by other robots within range (150 cm) to provide rough directional information to any robots in range. IR beacon signals are used at short range (15 cm) to guide the approaching robots for precise alignment with the docking face. Upon com-

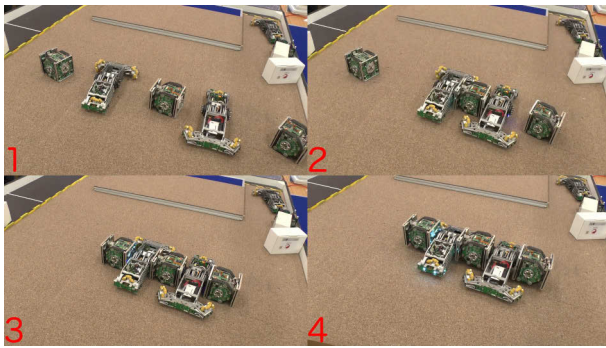


Figure 3: Morphogenesis in progress. Image 1: Five robots are in swarm mode. Image 2: Self-assembly is in progress. Image 3: The new organism is complete, but in 2D planar form. Image 4: The organism ‘stands up’ to transform to 3D.

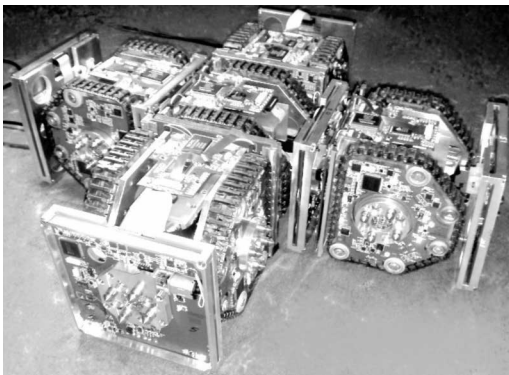


Figure 4: Example of a result from embodied morphogenesis using 5 Symbion robots obtained with the Virtual Embryogeny approach (credits: Markus Dauschan). See Dauschan et al. (2011) for details.

pletion of the docking process, robots stop emitting beacon signals. The same process is then repeated until the pre-evolved structure is formed. A behaviour-based approach is adopted for the design of the morphogenesis controller, together with a well-formatted tree structure which explicitly represents the organism body-plan, as described in Liu and Winfield (2012).

In this way robots initially form a 2D planar structure, see Figure 3 image 3. Once the robots in the 2D planar structure have assumed the correct functionality, according to their position in the body plan, the ‘organism’ will lift itself from 2D planar configuration to 3D configuration (as shown in Figure 3 image 4) and, with respect to locomotion, function as a macroscopic whole.

### Birth: Implicit Encoding for Morphogenesis

An alternative to direct encoding is to consider developmental and generative systems (or *implicit* encodings). In

this setup, the information contained in the genome encodes the *process* of construction rather than an explicitly formulated *plan* of construction. While developmental and generative systems have been studied for some time (cf. the works of Bentley and Kumar (1999); Stanley and Miikkulainen (2003); Bongard and Pfeifer (2003)), the very process of morphogenesis starting from a swarm of autonomous units and going towards a full assembled organism raise additional issues, as the actual morphogenesis should be considered as an embodied process: online and decentralized.

In the last five years, several approaches have been investigated in the Symbion project, from theoretical ideas to practical robotic implementations, as shown in Figure 4. These approaches have been explored and tested, either with simulated or real robots, and have investigated the benefits of deterministic vs. stochastic morphogenesis from different perspectives (either bio-inspired or completely artificial). On one side, genetic regulatory networks (GRN) and artificial ontogenic process have been considered (Thenius et al. (2010)). On the other side, cellular automata (CA) have been used to model the developmental process by considering each robot of the organism as a cell with a von Neumann neighbourhood. In both cases, cells would be considered as homogeneous, that is sharing the same evolved update rules, whether this was explicit CA rules, a GRN update network and any other kind of developmental program. However, each cell would then trigger the recruitment of other cells depending on their current (possibly unique) situation, ultimately leading to a full-grown organism having reached a stable final configuration, as explored by Devert et al. (2011).

What makes these approaches particular with respect to the literature is that it is not only necessary to encode the morphogenesis process itself (i.e. the assembling sequence), but it is also mandatory to consider the actual execution of this process (the *embodied* morphogenesis): individual units are indeed facing a possibly challenging coordination tasks, with the possible constraints of satisfying temporal and spatial constraints (e.g. the assembly ordering can be important). Moreover, open-ended evolution of embodied morphogenesis can benefit from a creative process, that is to come up with original morphological solutions to address the challenges in the environment at hand. We devised a set of performance indicators to encompass these various desired properties and these are described below.

*Evolvability* is considered as the ability for the algorithm to produce *viable* shapes during the course of evolution. It is evaluated by counting the number of unique viable shapes out of a predefined number of tries.

*Initial viability* provides an indicator to estimate how difficult it is to bootstrap an evolutionary process. It is computed by considering only random generations of genotypic description for the encoding under scrutiny, and counting the number of shapes that can actually be build (i.e. viable)

out of the total number of shape descriptions generated.

*Self-repair* stands as one of the typical benchmark for morphogenesis and evaluates how a full organism can be successfully reconstructed from a starting condition that may not match the original initial condition (e.g. from the last recruited robot rather than from the original "egg" robot).

Lastly, *controllability* (unsurprisingly) evaluates the efficiency with respect to evolving the construction process to achieve a particular target shape: the faster the evolution, the better the controllability.

### Infancy: Gait Learning

In our vision of Artificial Life based on hardware birth is followed by the stage of infancy. From an evolutionary point of view the proof of viability at the very beginning of this stage does not need any further consideration. If an organism, for example, consumes too much energy, its genome will not spread. Thus, in SYMBRION we concentrate on the objective of an organism learning to control its own body for locomotion. This is because movement increases the chances to spread the genome during the upcoming phase of mature life, independent of the chosen reproduction implementation during the mature phase. Thus, the objective of gait learning is an indirect one. The obvious easy solution of so-called free-riders, which are organisms staying in place and waiting for others to come by, can only exist in a low number in the population from an evolutionary perspective.

Here, gait learning comes with challenge of an unknown body shape. There may have been passed on a genome performing good locomotion from the ancestor but this good performance does not automatically hold for a different body shape. Thus, investigations on gait learning for a modular multi-robot organism –as it is the case in SYMBRION– always start from scratch.

As mentioned above on-line, on-board evolution was chosen in SYMBRION to be the optimization process. This leads to several important consideration and scientific questions. For example, the part of the genome which is responsible for locomotion could use Lamarckism. This means that at the beginning of mature life not the original genome but the genome altered by artificial evolution during gait learning is used for recombination.

The way of achieving shared control is another consideration. Should the controllers of the single modules in the organism be derived from an identical genome ("homogeneous")? Different genomes ("heterogeneous") would ease the creation of division of labor as some cells would be used to push, others to pull. In Waibel et al. (2009) it is stated that a homogeneous genome of team members is better suited when the task requires high level of cooperation.

Another important aspect is the type of controller being used. This strongly depends on the actuators used for locomotion. The three robot platforms which are the modules of



Figure 5: Multi-robot organism consisting of three modules during infancy. Screenshots show attempts of the organism to create locomotion during on-line, on-board evolutionary.

the multi-robot organism in SYMBRION come with several 2D actuators and one 3D actuator. The primary focus has been on the 3D drive. It is implemented as hinges which makes it possible to lift the other modules. Fig. 5 shows an example of the resulting 3D locomotion of an organism. This leads to a snake- or caterpillar-like motion. Three different controller types known for their evolvability were taken into consideration: CPG (central pattern generator), AHHS (artificial homeostatic hormone system, see Stradner et al. (2012)) and GRN (gene regulatory network). The idea is not to limit the population to one solution in the first place but to let evolution decide. The organism will only be controlled by one type during infancy phase, but the better it performs the greater the chance that this type will also be used by its offspring.

The ongoing work in SYMBRION is the implementation and testing (first results are shown in Fig. 5) for experiments to investigate the considerations raised above concerning gait learning for multi-robot organisms.

### Mature Life: Self-Reproduction

Weel et al. (2013) recently described an egg-based system extending the seed-based protocol from the previous section. The idea is that some of the robot modules that are not part of a robot organism act as an egg whose function is to collect and process genomes of robot organisms for reproduc-



tion. An egg is thus a stationary robot module that organisms can fertilize by sending their genome to it. An egg that has been fertilized by a number of organisms selects two of the received genomes for recombination followed by mutation. Then the egg becomes a seed, and initiates the morphogenesis of a new organism using the new genome.

This system has been implemented in a rather simple, fast simulator, RoboRobo<sup>3</sup> and numerous experiments have been conducted to gain insights into the ‘inner logic’ of this system. In particular, three major parameters have been identified: *egg lifetime*, i.e., how long eggs listen for genomes, *seed lifetime*, i.e., how long a fertilized egg (a seed) is allowed to build the organism its genome encodes before aborting, and *organism lifetime*, i.e., how long a fully grown organism lives before it dies. These experiments have disclosed how these parameters interact, in particular regarding their influence on the size of the organism population, the stability of the organism population, and the average size of the organisms.

### Mature Life: Self-Repair

There are many complex steps proposed in this paper: birth, infancy, mature life over a sustained period. All of these complex and potentially error prone steps may well cause, or be inhibited by faults. Hence, throughout the lifetime of the robotic system, it is inevitable that there will be some form of failure within a robot, or within the organism. When such failures occur, the ability of the organism to perform its task, or even survive, is compromised. Failures can be caused by a range of different faults ranging from mechanical failures, to electronic hardware or software faults and as such prevent the organism from performing its task. For continued operation over the full lifetime of the robot/organism some form of self-repair is needed. The ability to self-repair is linked to the ability of the organism to perform morphogenesis, as it is very likely that some form of reconfiguration is needed in the event of failure.

We report here on two approaches of self-repair that have been explored. The first could be considered a type of self-assembly, as reported in Murray et al. (2013) where robots are able to form ad-hoc structures, with no pre-determined shape, as opposed to work described above where a shape is seeded into the robotic unit. Murray et al presented an algorithm that showed successful reconfiguration ability of specifically tailored e-pucks that could form the aforementioned structures.

Further work by the SYMBRION project, as yet unpublished, goes much further to permit a true self-repair approach for organisms. Using techniques developed within the project for the detection Timmis et al. (2010) and diagnosis Bi et al. (2010) of faults, combined with the morphogenesis approach described here, SYMBRION organisms can

perform a partial disassembly then a full reassembly back to the original structure, in a distributed and autonomous manner. Should a robotic unit fail at any position within the organism, the approach permits for the removal of that unit and a reconstruction of the organism using the morphogenesis approach described.

### Concluding Remarks

In this paper we have introduced the Triangle of Life: a conceptual framework for artificial systems in which robots actually reproduce. Our proposed framework contrasts with traditional evolutionary robots approaches in several ways. Firstly, the life cycle does not run from birth to death, but from conception (being conceived) to conception (conceiving one or more children). Secondly we envision the whole process taking place in real time, with real robots in the real world. We do not prescribe how the process should be implemented, but two contrasting approaches present themselves: one in which some infrastructure provides materials and processes for robot birth, and another infrastructure-less approach which could be thought of as an extension to modular self-assembling robotics. The third departure from conventional practice is that fitness is tested primarily through survival to maturity and successful mating, rather than against an explicit fitness function. Thus a large number of factors including individual health and development, the living environment (which may include multiple generations of conspecifics), and simple contingency will influence whether an individual survives to pass on its genetic material. Importantly it follows that selection is also implicit. Although we are describing an artificial life system, the process of selection is much closer to Darwinian natural selection.

Finally we should speculate on how such an artificial life system might be used. Two contrasting applications present themselves. One as an engineering solution to a requirement for multiple robots in extreme unknown or dynamic environments in which the robots cannot be specified beforehand: robots required to explore and mine asteroids, for instance. The other application is scientific. Our proposed artificial life system could be used to investigate novel evolutionary processes, not so much to model biological evolution – life as it is, but instead to study life as it could be.

### Acknowledgements

The authors gratefully acknowledge the contributions of all SYMBRION partners. In particular, we are indebted to Markus Dauschan, Evert Haasdijk, Serge Kernbach, Wenguo Liu, Jean-Marc Montanier, Jessica Meyer, Lachlan Murray, Marc Schoenauer, Michele Sebag, and Berend Weel.

### References

- Bentley, P. and Kumar, S. (1999). Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In et al., W. B., editor, *GECCO'99*, pages 35–43. Morgan Kaufmann.

<sup>3</sup><https://code.google.com/p/roborobo/>



- Bi, R., Timmis, J., and Tyrrell, A. (2010). The diagnostic dendritic cell algorithm for robotic systems. In *12th IEEE Congress on Evolutionary Computation (CEC10)*, pages 4280–4287.
- Bongard, J. and Pfeifer, R. (2003). Evolving complete agents using artificial ontogeny. In Hara, F. and Pfeifer, R., editors, *Morpho-functional Machines: The New Species (Designing Embodied Intelligence)*, pages 237–258. Springer-Verlag, Berlin.
- Dauschan, M., Thenius, R., Schmickl, T., and Crailsheim, K. (2011). Using virtual embryogenesis multi-robot organisms. In *International Conference on Adaptive and Intelligent Systems, ICAIS'11, Klagenfurt, AT, September 06-08, 2011. Proceedings*, pages 238–247.
- Devert, A., Bredeche, N., and Schoenauer, M. (2006). Blind-builder: a new encoding to evolve lego-like structures. In Collet, P., Tomassini, M., Ebner, M., Gustafson, S., and Ekart, A., editors, *Proceedings of EUROGP 2006*, pages 61–72. Springer.
- Devert, A., Bredeche, N., and Schoenauer, M. (2011). Robustness and the halting problem for multicellular artificial ontogeny. *IEEE Trans. Evolutionary Computation*, 15(3):387–404.
- Doursat, R., Sayama, H., and Michel, O., editors (2012). *Morphogenetic Engineering: Toward Programmable Complex Systems*. Springer series: Understanding Complex Systems. Springer.
- Eiben, A.E., Kernbach, S., and Haasdijk, E. (2012). Embodied artificial evolution – artificial evolutionary systems in the 21st century. *Evolutionary Intelligence*, 5(4):261–272.
- Funes, P. and Pollack, J. (1997). Computer evolution of buildable objects. In Husbands, P. and Harvey, I., editors, *Fourth European Conference on Artificial Life*, pages 358–367. MIT Press.
- Langdon, C., editor (1989). *Proceedings of the Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (ALIFE '87)*. Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley.
- Langton, C., editor (1995). *Artificial Life: an Overview*. MIT Press, Cambridge, MA.
- Levi, P. and Kernbach, S., editors (2010). *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*, volume 7 of *Cognitive Systems Monographs*. Springer, Berlin, Heidelberg, New York.
- Levy, S. (1992). *Artificial Life*. Vintage Books.
- Lipson, H. and Pollack, J. B. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, 406:974–978.
- Liu, W. and Winfield, A. F. (2012). Distributed autonomous morphogenesis in a self-assembling robotic system. In Doursat, R., Sayama, H., and Michel, O., editors, *Morphogenetic Engineering*, Understanding Complex Systems, pages 89–113. Springer Berlin Heidelberg.
- Lund, H. (2003). Co-evolving control and morphology with lego robots. In *Morpho-functional Machines: The New Species*, pages 59–79. Springer.
- Murata, S. and Kurokawa, H. (2012). *Self-Organizing Robots*, volume 77 of *Springer Tracts in Advanced Robotics*. Springer.
- Murray, L., Timmis, J., and Tyrrell, A. (2013). Modular self-assembling and self-reconfiguring epucks. *Swarm Intelligence*. (in press).
- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA.
- Peysakhov, M., Galinskaya, V., and Regli, W. C. (2000). Using graph grammars and genetic algorithms to represent and evolve lego assemblies. In *Proceedings of GECCO 2000*, pages 269–275.
- Stanley, K. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.
- Stradner, J., Hamann, H., Zahadat, P., Schmickl, T., and Crailsheim, K. (2012). On-line, on-board evolution of reaction-diffusion control for self-adaptation. In Adami, C., Bryson, D. M., Ofria, C., and Pennock, R. T., editors, *Proceedings of the ALife13*, pages 597–598. MIT Press.
- Thenius, R., Bodi, M., Schmickl, T., and Crailsheim, K. (2010). Using virtual embryogenesis for structuring controllers. In Hart, E., McEwan, C., Timmis, J., and Hone, A., editors, *Artificial Immune Systems, 9th International Conference, ICARIS 2010*, volume 6209 of *Lecture Notes in Computer Science*, pages 312–313. Springer.
- Timmis, J., Tyrrell, A., Mokhtar, M., Ismail, A., Owens, N., and Bi, R. (2010). An artificial immune system for robot organisms. In Levi, P. and Kernbach, S., editors, *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*, pages 268–288. Springer.
- Trianni, V. (2008). *Evolutionary Swarm Robotics – Evolving Self-Organising Behaviours in Groups of Autonomous Robots*, volume 108 of *Studies in Computational Intelligence*. Springer.
- Waibel, M., Keller, L., and Floreano, D. (2009). Genetic Team Composition and Level of Selection in the Evolution of Cooperation. *IEEE Transactions on Evolutionary Computation*, 13(3):648–660.
- Wang, L., Tan, K., and Chew, C. (2006). *Evolutionary Robotics: from Algorithms to Implementations*, volume 28 of *World Scientific Series in Robotics and Intelligent Systems*. World Scientific.
- Weel, B., Haasdijk, E., and Eiben, A.E. (2013). Body building: Hatching robot organisms. In *IEEE International Conference on Evolvable Systems (IEEE ICES 2013)*. IEEE Press.
- Yim, M., Shen, W., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., and Chirikjian, G. (2007). Modular self-reconfigurable robot systems. *IEEE Robotics and Automation Magazine*, 14(1):43–52.
- Zykov, V., Mytilinaios, E., Desnoyer, M., and Lipson, H. (2007). Evolved and designed self-reproducing modular robotics. *IEEE Transactions on Robotics*, 23(2):308–319.

## Radiofrequency triggered enzymatic reaction inside hydrogel microparticles

Martin Ullrich<sup>1</sup>, Petra Haufová<sup>1</sup>, Mandeep Singh<sup>1</sup> and František Štěpánek<sup>1</sup>

<sup>1</sup> Department of Chemical Engineering, ICT Prague, Technická 5, Prague 6 – Dejvice, 166 28, Czech Republic,  
tel: +420 220 443 048  
Martin.Ullrich@vscht.cz

### Introduction

Biochemical reactions in living organism usually take place inside cells. The characteristic feature of these reactions is their executability; the organism does not produce certain substances all the time, but only when they are needed. The reaction is usually triggered after receiving an extra- (or intra-) cellular signal. Stimuli-responsiveness is mostly possible because of compartmentalization of the cells. Reactants, enzymes and other factors are separated to different parts of cells by membranes forming organelles. Upon receiving a signal, the substances are selectively transported through the membranes and a reaction occurs.

In order to mimic that behaviour we present internally-structured cell-like micro-particles with a diameter around 50  $\mu\text{m}$ . The particles are formed by an alginate gel, thermo-responsive liposomes form internal compartments filled by an encapsulated content. The particles also contain ferrofluid ( $\text{Fe}_2\text{O}_3$  magnetic nanoparticles). High-frequency magnetic field causes heating of the particles hence release the content from thermo-sensitive liposomes into the gel body and then out of the particles (Hanuš, et al. 2013). Thermo-sensitive liposomes are characteristic by a change of bilayer structure at certain temperature.

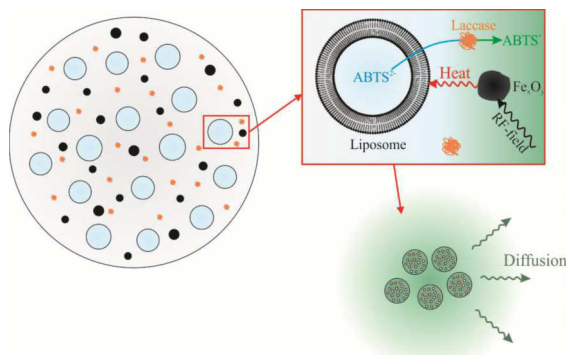


Figure 1: Scheme of the particle and principle of triggering the reaction. The substrate is released from liposomes after induction heating of particles and it is oxidized in the gel by means of the immobilized enzyme.

Utilizing these particles we studied an enzymatic oxidation of ABTS (2,2'-azino-bis(3-ethylbenzothiazoline-6-sulphonic

acid)) catalyzed by the enzyme laccase. The substrate was encapsulated inside the liposomes and the enzyme was immobilized into the alginate hydrogel. A scheme of the particle and the principle of reaction triggering is shown in Figure 1.

### Preparation

Liposomes were made from DPPC (1,2-dipalmitoyl-*sn*-glycero-3-phosphocholine) and cholesterol at a molar ratio of 8:1 (DPPC:cholesterol). They were prepared by the hydration method followed by an extrusion through 100 nm polycarbonate membrane. Magnetic nanoparticles were made by the precipitation of  $\text{FeCl}_2$  and  $\text{FeCl}_3$  in an ammonia solution. A mixture of the ferrofluid, liposomes with ABTS, laccase and sodium alginate was dripped into a solution of copper(II) sulphate by a syringe with a needle in order to produce 1-2mm particles (Ullrich, et al. 2013) or by an ink-jet print-head in the case of micro-particles ( $\sim 80 \mu\text{m}$ ) (Haufová, et al. 2012). Microscopic image of such micro-particles is in Figure 2. After 1 hour of cross-linking, the particles were washed with water and then transferred to 10 mM acetate buffer (pH = 5.0). The product of the enzymatic reaction is intensively coloured; this enables spectroscopic detection at 415 nm.

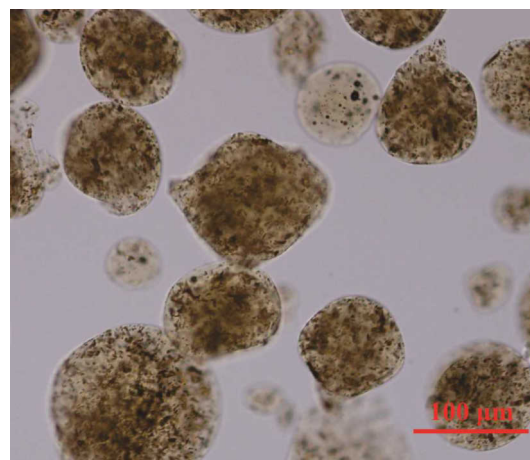


Figure 2: Microscopic image of ink-jetted micro-particles containing iron oxide nanoparticles, laccase and liposomes enclosing solution of ABTS.

## Results

At first, laccase was immobilized within calcium alginate gel. A significant decrease of the enzyme activity was observed. Leakage of the enzyme during cross-linking and the presence of calcium ions was found to have the main effect on the enzyme deactivation. After 1 hour of gel cross-linking, about half of the original amount of laccase leaked out of the beads. Moreover, the activity of laccase in 150 mM  $\text{CaCl}_2$  was about 10-times lower than in 100 mM acetate buffer. On the other hand, the enzymatic reaction was found to be slightly faster in 150 mM copper(II) sulphate compared to the buffer. That led us to use copper sulphate solution for cross-linking instead of calcium chloride that was used originally for the preparation.

The temperature-dependent release rate of the product has been investigated. At first, millimetre-sized particles were studied. It was found that approximately 10 % of the substrate was released at room temperature ( $\sim 23^\circ\text{C}$ ) in 3 hours, but the leakage could be decreased to 1 % by storage in a refrigerator. On the other hand, only 5 minutes in a  $50^\circ\text{C}$  water bath was enough to release and oxidize almost all of the encapsulated substrate.

Similar experiment was carried out with iron-oxide nanoparticles and RF-heating by an induction coil (frequency 400 kHz, max. amplitude 20 mT  $\sim 100\%$  in Figure 3). The effect of the magnetic field intensity and duration of the RF-pulse were studied. Precise control of the magnetic field led to a step-wise release hence dosage control of the product. Functionality of this principle was then verified on micrometre-sized particles prepared by the ink-jet printing (Figure 3).

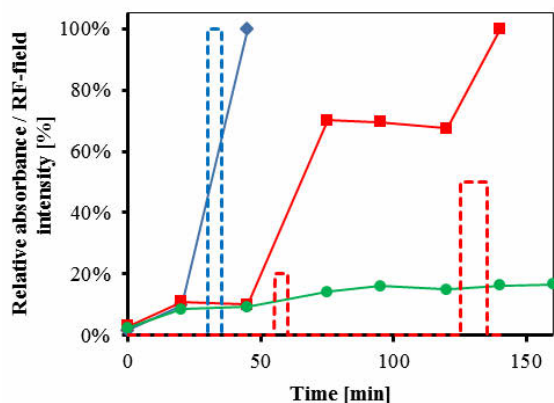


Figure 3: Release of the reaction product from alginate micro-particles. Legend: (♦)(■) samples heated by an induction coil; (●) not-heated reference sample; (- -) intensity of RF-heating. Relative absorbance 100% refers to an absorbance after release all the product.

## Conclusions

Artificial cell-like particles have been created. Liposomes have a function of inner compartments and an alginate gel forms an artificial protoplasm. The particles contain also iron oxide nanoparticles enabling a control of processes inside by a temperature change. Utilizing this system we demonstrated the

ability to control the rate of reaction-diffusion processes remotely by a RF-field. Micro-particles containing thermo-sensitive liposomes, holding the substrate for enzymatic reaction (ABTS), and an immobilized enzyme (laccase) released a significant amount of the reaction product after an exposure to a RF-magnetic field. This opens up the possibility to use such particles as micrometre-scaled bioreactors for more complex reaction networks using liposomes with different composition hence a different bilayer transition temperature. Also local, on-demand synthesis and release of pharmacologically or otherwise active compounds with a short lifetime could be possible.

## References

- Hanuš, J., Ullrich, M., Dohnal, J., Singh, M., Štěpánek, F. (2013). Remotely Controlled Diffusion from Magnetic Liposome Microgels. *Langmuir*, 29(13):4381–4387.
- Haufová, P., Dohnal, J., Hanuš, J., Štěpánek, F. (2012). Towards the inkjet fabrication of artificial cells. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 410:52–58.
- Ullrich, M., Hanuš, J., Dohnal, J., Štěpánek, F. (2013). Encapsulation stability and temperature-dependent release kinetics from hydrogel-immobilised liposomes. *Journal of Colloid and Interface Science*, 394:380–385.

ECAL 2013 Submission 37, Evolvable Hardware, Evolutionary Electronics & BioChips Track.

Linking Evolution in Silico, Hardware, and Chemistry to discover or engineer Inorganic Biology

Leroy Cronin, School of Chemistry, WestCHEM, The University of Glasgow, Glasgow, G12 8QQ, UK.

Email: [Lee.cronin@glasgow.ac.uk](mailto:Lee.cronin@glasgow.ac.uk); web: <http://www.croninlab.com>

Keywords: inorganic biology; 3d printers; artificial life; inorganic chemical cells (iCHELLs); 3d printer-based liquid handling robots; flow-bots

In our laboratory we have been developing new approaches to discover the 'transition-to-evolvability' in chemistry. This is because if we can discover or engineer an abiotic system that can evolve (we could define this as an inorganic chemical cell –iCHELL, see Figure)[1] we might be able to suggest that synthetic biology can exist in many chemical forms, of which the terrestrial biology found on planet earth is one subset. It could even help us establish the idea that evolvability is the key signature that defines living from non-living systems. In this contribution I will describe how we are connecting evolutionary algorithms, hardware (e.g. flow systems [2], 3d printers[3] and liquid handling robots) and complex chemical systems to produce new types system without classically defined biological genetic material, yet with the potential to evolve.

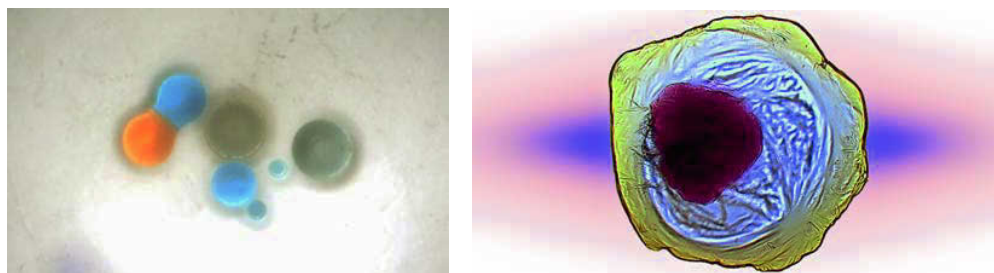


Figure: LEFT picture of some CHELLS fusing. RIGHT picture of an iCHELL made of inorganic salt with an inner 'metabolic' payload.

[1]. G. J. T. Cooper, P. J. Kitson, R. Winter, M. Zagnoni, D.-L. Long and L. Cronin 'Modular Redox Active Inorganic Chemical Cells: iCHELLs', *Angew. Chem. Int. Ed.*, 2011, 50, 10373. DOI: 10.1002/anie.201105068

[2]. C. J. Richmond, H. N. Miras, A. R. de la Oliva, H. Zang, V. Sans, L. Paramonov, C. Makatsoris, R. Inglis, E. K. Brechin, D-L. Long, L. Cronin, 'A flow-system array for the discovery and scale up of inorganic clusters' *Nature Chem.*, 2012, 4, 1037-1043. DOI:10.1038/nchem.1489

[3]. M. D. Symes, P. J. Kitson, J. Yan, C. J. Richmond, G. J. T. Cooper, R. W. Bowman, T. Vilbrandt, L. Cronin, 'Integrated 3Dprinted reactionware for chemical synthesis and analysis', *Nature Chem.*, 2012, 4, 349-354. DOI :10.1038/nchem.1313



# Evolution of an artificial visual cortex for image recognition

Samuel D. Chapman<sup>1,3</sup>, David B. Knoester<sup>2,3</sup>, Arend Hintze<sup>2,3</sup>, and Christoph Adami<sup>2,3</sup>

<sup>1</sup>Dept. of Computer Science and Engineering

<sup>2</sup>Dept. of Microbiology and Molecular Genetics

<sup>3</sup>BEACON Center for the Study of Evolution in Action

Michigan State University, East Lansing, MI 48824

Email: {chapm236, dk, hintze, adami}@msu.edu

## Abstract

Evolutionary robotics has been successful in creating agents that successfully link perception with appropriate action. However, the visual fields utilized by such agents is usually extremely small compared to the retinas linked to the visual cortex of animals. Evolving a cortex that processes larger fields of view in a selective and robust manner is challenging because fitness landscapes that are sensitive to this level of details are difficult to design. Here, we decouple the perception from the action part of evolutionary robotics, and present a new way to evolve logic circuits to perform image recognition on the well-known MNIST data set, which comprises 60,000 training and 10,000 testing handwritten numerals. The logic circuits are encoded in a genome that is evolved using a fitness function based on the true positive and true negative classification rates of the numerals. Following evolution, individual circuits achieve in excess of 80% recognition accuracy on the testing data. By pooling highly evolved individual circuits from multiple evolutionary histories into a committee, testing accuracy is increased to 93.5%.

This work demonstrates that evolving logic circuits to solve a classification task is feasible. We also found the evolved circuits to be much smaller in scale compared to other machine learning methods that are conventionally used on such problems. To our knowledge, this represents the first time that relatively small logic circuits have been evolved to reach this level of performance on the automated recognition of handwriting, and promises new approaches to the integration of evolutionary algorithms and intelligent systems.

## Introduction

The primary visual cortex of mammals is a marvel of complexity and function, and has been studied for decades both by those wishing to *understand* it, as well as those wishing to *duplicate* it. Object recognition by the visual cortex is fast (Thorpe et al., 1996), specific, and invariant: we can recognize faces, for example, in different orientations, under a variety of lighting conditions, and bearing different expressions. Significant efforts have been expended on modeling the hierarchical organization of the visual cortex, with the aim of both elucidating the algorithm behind fast and invariant object recognition (Riesenhuber and Poggio, 1999), as well as harnessing it for computer vision (Serre et al., 2007).

The usefulness of such a system for the organism bearing it is almost too obvious to point out: it is clear that the selective advantage of a fast, accurate, and robust sensing system must be enormous. Indeed, the visual cortex evolved to be the largest system in the human brain.

While significant progress has been made on our understanding of the visual cortex, the development of a visual cortex suitable for embodied robotics (Pfeifer and Bongard, 2006) remains a challenging problem. Typically, embodied mobile robots need to carry lightweight, fast, and energy-efficient components, and typically such systems have a very limited bandwidth. Most (if not all) of the implementations of automatic computer vision fall short in one or more of these requirements. To be sure, advanced systems such as Google's self-driving car (Ibañez-Guzmán et al., 2012) have achieved extremely robust object recognition and navigation skills, however at the cost of processing almost a Gigabyte/s of sensorial data (Chau et al., 2013). Less advanced systems (such as, say, an ant) navigate robustly with arguably nine orders of magnitude less sensorial data, using a far more compact circuitry, and minimal energy requirements. How can we understand the structure and function of such visual systems?

## Evolutionary robotics

Within the field of evolutionary robotics (Nolfi and Floreano, 2000; Floreano and Keller, 2010), it is becoming more and more clear that algorithms for perception should not be separated from those for action, that instead perception and action should be considered as an integrated system computing appropriate behavior from contextual information. Evolving the neural control structures for navigation (Floreano and Mondada, 1998; Edlund et al., 2011) has the advantage that it is free of any preconceived notions of what is designable and the added bonus of robustness, because the prime directive of an evolutionary fitness landscape is the survival of the agent. Advances have been made specifically in the area of *active categorical perception*, where an agent actively moves to better categorize the object it sees (Beer, 1996, 2003; van Dartel et al., 2005; Marstaller et al., 2013).

In this type of work, the visual system of the agent is necessarily primitive, as the emphasis lies squarely on the evolution of the perception-action loop. One approach to extend this work could be to simply extend the size of an agent's retina, with the hope that the added detail within the visual field will prove useful for robot behavior. The drawback of this approach is familiar to most that have worked in this field: designing a fitness landscape (Nelson et al., 2009) that makes exquisitely accurate and robust image recognition necessary is extremely challenging, because in most artificial worlds an agent can achieve success with relatively few cues. Floreano et al. (2005) were able to evolve robots that navigate using an evolved retina with a resolution of  $5 \times 5$  visual neurons, and showed that selection tried to exploit a subset of salient visual features, rather than to process the whole available image.

In order to test whether image recognition algorithms can be evolved in the absence of actions—and to circumvent the problem of having to design a fitness landscape where fast, robust and selective image recognition is necessary for appropriate behavior—we decided to study whether evolution can re-create an object-recognition system suitable for embodied robotics, by evolving an artificial visual cortex that recognizes hand-written numerals. This task (a sub-problem of what is commonly referred to as “optical character recognition”, or OCR) is a well-known staple of the machine learning community, and as a consequence several benchmarks are available to compare performance. In our work (as opposed to the standard machine learning application), we have more goals besides trying to achieve maximum accuracy in image classification. We also require the resulting network to be fast, small, and easily transferrable from one computing platform to another. But most importantly, the solution must be evolvable. We can imagine, for example, that the evolved visual cortex would be connected to the neural controlling machinery of an embodied robot, so that both can then evolve together.

In what follows, we use evolutionary algorithms to evolve logic circuits that classify the numerals from the MNIST data set (LeCun et al., 1998), a well-known benchmark in OCR. Evolutionary algorithms employ the principles of evolution by natural selection in a computational context to solve complex problems (Bäck, 1996). Our approach makes use of a framework to evolve *Markov networks* (MN), a development from recent work (Edlund et al., 2011). MNs comprise a series of state variables that interact with one another through a collection of evolved “computational gates.” These gates specify how state variables change in response to one another. Gates can be either probabilistic, much like fuzzy logic, or deterministic, as in traditional logic circuits. The Markov networks themselves are represented by a circular list of integers (the “genome”) that describes each gate and the state variables with which they interact. Mutations within our evolutionary algorithm operate directly on these

genomes, and we employ a fitness function that is based on true positive and true negative recognition rates. In this work, we make use of deterministic gates only, and thus evolve logic circuits that are essentially complex boolean functions that classify numerals from the MNIST data set.

Previous techniques applied to OCR on handwritten characters include K-nearest-neighbors (LeCun et al., 1998), support vector machines (Decoste and Schölkopf, 2002), and learning classifiers such as artificial neural nets and convolutional nets (LeCun et al., 1998; Ciresan et al., 2012; Salakhutdinov and Hinton, 2007). In the case of neural network-based approaches, edge weights are typically optimized based on an error-minimization algorithm such as backpropagation. However, the number of network layers, along with the number of computational units per layer, remains fixed during training. Over time, these approaches have improved in performance on the MNIST data set. For example, in 1998 the best neural network *error rate*, the rate of misclassification on the test images, was 1.6%, achieved using a 2-layer neural network with 300 hidden states (LeCun et al., 1998). In 2007, Salakhutdinov and Hinton achieved an error rate of 1.0%, using a “deep learning” neural network architecture (Salakhutdinov and Hinton, 2007). Deep learning involves the use of a higher-than-normal number of layers; in this case, five neural network layers were used, with all but the last layer using hundreds of nodes. In 2010, Ciresan *et al.* were able to decrease the error rate to 0.35%, using six layers and an even greater number of nodes per layer (Ciresan et al., 2010). The current world record error rate for the MNIST data set is 0.23%, again set by Ciresan *et al.* in 2012, where they used a 35-member committee (7,700 total neurons) of convolutional networks trained via back-propagation for 490 hours on GPU (Ciresan et al., 2012).

Here we use evolution to discover logic circuits that classify the numerals from the MNIST data set. The evolved circuits are of larger scale than other evolutionary approaches, having 784 inputs and 20 outputs, and unlike ANNs and convolutional networks, evolve to a highly concise representation of about 100 computational gates. While individual evolved logic circuits have remarkable accuracy given their relatively small size, the diversity inherent to evolutionary algorithms enables us to make effective use of *committees*. Using committees of 30 circuits, we achieve an accuracy of approximately 93.5% on the MNIST data set. This level of scalability of logic circuit evolution has not been seen in other studies (Stomeo et al., 2005; Vassilev and Miller, 2000; Torresen, 2001), and demonstrates that the evolution of large logic circuits to perform a practical application is feasible. Finally, because Markov networks represent logic circuits, they are capable of being rendered on physical hardware such as FPGAs.

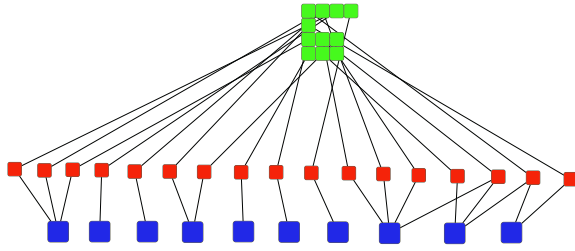


Figure 1: A schematic showing how a Markov network operates. Inputs (green) in a 4x4 image field connect to logic gates (red), which produce the 10 outputs (blue).

## Methods

### Evolving logic circuits

Defined as a set of probabilistically interacting state variables (Koller and Friedman, 2009), Markov networks (MNs)—which are frequently used to model stochastic processes—can also encode a wide variety of behaviors. The state variables (SVs) represent inputs to the MN, outputs from the MN, and “hidden state variables,” that is, SVs that are internal to the MN. A proof-of-principle that MNs can successfully evolved to control behavior (including the usage of memory) was provided in a study of animat navigation in a maze (Edlund et al., 2011). There, some SVs act as sensors of the external environment, and other SVs act as motor outputs and memory. To evolve logic circuits, we use that framework for the evolution of MNs, however, here we use only deterministic gates, as we will discuss in more detail below. Figure 1 depicts an example Markov network. Here, inputs to the MN are pixels from a small 4x4 image (green). These inputs are provided to a series of logic gates (red), which in turn produce outputs (blue).

State variables are connected to each other in a directed fashion in a MN, and this connection is mediated through logic gates. Each gate is connected to a number of SVs as input and produces a specified number of SVs as output. During each network update, the inputs to each gate are used to calculate the values that will be written into the gate’s output SVs. The outputs produced by a gate thus depend on the value of the input SVs and the specific logic of the gate. To illustrate how these gates work, Table 1 shows a truth table for a 2-input, 2-output logic gate. Here, this gate is connected to input SVs  $a$  and  $b$  and output SVs  $c$  and  $d$ . If  $a$  and  $b$  take the binary values 1 and 0 respectively during one *tick* (a tick represents one execution of all logic gates in a MN), then this gate will write a 1 into both output SVs  $c$  and  $d$ . If more than one gate writes into the same SV, the values are combined via a logical OR.

To evolve logic circuits, we encode the circuit within a digital genome: a circular list of integers. Within this genome, each gate is encoded by a gene. The beginning

Table 1: A sample logic table for a deterministic gate with 2 inputs and 2 outputs.

| Inputs |     | Outputs |     |
|--------|-----|---------|-----|
| $a$    | $b$ | $c$     | $d$ |
| 1      | 1   | 0       | 1   |
| 0      | 1   | 0       | 0   |
| 1      | 0   | 1       | 1   |
| 0      | 0   | 0       | 1   |

of each gene is identified by a specific marker (the “start codon”), and the gene defines the entire functionality of its associated gate. Specifically, genes specify the identity of input and output state variables for each gate, as well as the logic table defining the operation of the gate. In this study, the genome is limited to 40,000 integers, which has an overall capacity of a few thousand genes.

In the evolutionary process, the genome is subject to point mutation, duplication, or deletion. A *point mutation* in the genome randomly changes the value of an integer to another value, whereas a *duplication* or *deletion* inserts random integers into or removes a small section from the genome, respectively. Aside from selecting the range from which random integers are drawn, there are no constraints placed on the values of the integers within the genome. For this reason, not all genes will necessarily be useful toward the problem at hand. Of course, as with natural genomes, nonfunctional genes can serve as a “bank” of genetic diversity that evolution can make use of. Finally, while the MN framework allows the input and output dimension of gates to change via mutation, for simplicity here we fix gates to 4 inputs and 4 outputs. A detailed exposition of the encoding of gates within the circular list of integers is provided in the supplementary information of Edlund et al. (2011).

### Recognizing characters

Each image in the MNIST data set is rendered as a  $28 \times 28$  pixel grayscale image, where each pixel has a grayscale value from 0 to 255. A sample of these images is shown in Fig. 2, along with a sample of difficult to classify images on the right. In the literature, numerous pre-processing steps are often taken in order to render the images more conducive to particular machine-learning methods, but here we use a simple binary transform on the raw images as delivered in a previous study (LeCun et al., 1998). Specifically, we transform each image such that all pixels with a gray level of 0 are treated as a binary “0,” and all nonzero pixels are treated as a binary “1.” Each of the 784 binary pixel values are then provided to MNs as inputs. Each MN produces 20 outputs, which represent the classification produced by the MN. Each pair of output bits represents a yes/no answer for one class. Specifically,  $c_i = b_{0,i} \wedge \neg b_{1,i}$ , where  $b_{0,i}$  is the first bit for class  $i$ ,  $b_{1,i}$  is the second bit for class  $i$ , and

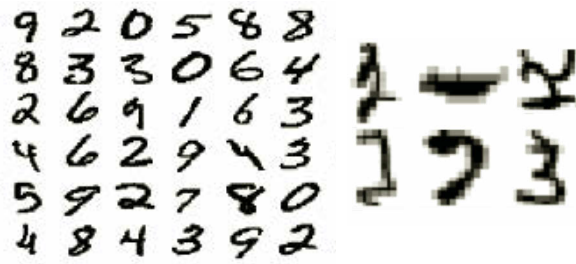


Figure 2: A sample of grey-scale images of hand-written numerals from the MNIST database (left), along with a set of difficult to classify images from the same set.

$c_i$  is the MNs boolean decision for whether the image under consideration is thought to belong to class  $i$ .  $b_{0,i}$  can be thought of as the activating bit for a class decision, whereas  $b_{1,i}$  serves an inhibitory function such that it negates a positive class decision by  $b_{0,i}$ . We allow the MNs to produce multiple answers for each image. This may seem counter-intuitive, but with the appropriate choice of fitness function, the MNs can evolve to only guess the correct answer, while producing negative answers for the other classes.

The fitness function used to evolve MNs is based on the true positive and true negative rates (TPR and TNR, respectively) of a network's decisions for each class. In essence, a network is rewarded for identifying which class an image belonged to, and also for identifying which classes it did *not* belong to. If even one of these details was correct, the network receives some fitness for that guess. By allowing such "partial credit," we provide a more smooth fitness landscape than if the exact answers for an image were to be required. The specific fitness function we use is:

$$f = \sqrt{\sum_{i=0}^9 (\text{tpr}_i + \text{tnr}_i + \text{out}_i)^2}, \quad (1)$$

where  $\text{tpr}_i$  is the measured true positive rate on class  $i$ ,  $\text{tnr}_i$  is the true negative rate on class  $i$ , and  $f$  is the resulting fitness. If the network outputs a positive decision for *any* image of class  $i$ ,  $\text{out}_i$  is set to 1.0, thus encouraging the networks to evolve to classify *all* numerals (as opposed to simply focusing on the more easily classified numerals). Ties (outputting multiple classifications for a single image), are broken randomly to produce an accuracy score. We did not use the accuracy score of the network as fitness, as the resulting landscape proved too difficult to adapt to (data not shown). Note that when reporting fitness, we show the percentage of the maximum fitness achievable via Eq. (1).

The MNIST data set is divided into two parts: A set of 60,000 training images, and a set of 10,000 testing images. The Markov networks were only evolved on the training images. After evolution on the training set, the most fit individual network was then tested on the testing set. The accuracy

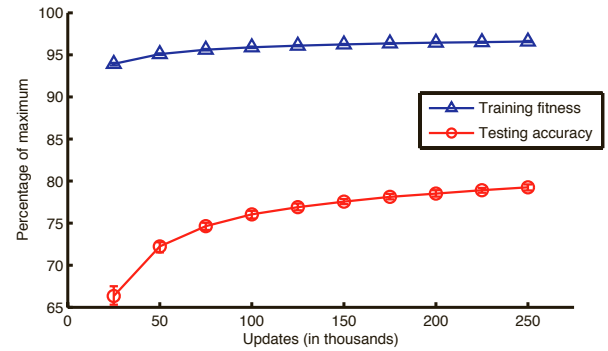


Figure 3: Mean training fitness and test accuracy of dominant individuals over time, over 30 independent runs. Using 200-fold bootstrapping, (small) error bars are constructed depicting 95% confidence intervals around the mean.

score of this individual was calculated as the fraction of test images correctly identified by the circuit. The parameters of our evolutionary algorithm are summarized in Table 2. In all, 30 replicate populations were evolved on the training set. After every 25,000 updates (up to 250,000 updates), we tested the highest-fitness individual from each replicate and recorded its accuracy.

Table 2: Parameters for the evolutionary runs.

| Parameter                        | Value   |
|----------------------------------|---------|
| Updates                          | 250,000 |
| Population size                  | 500     |
| Starting gates                   | 100     |
| Max inputs                       | 784     |
| Max outputs                      | 20      |
| Inputs per gate                  | 4       |
| Outputs per gate                 | 4       |
| Gene duplication rate per update | 0.05    |
| Gene deletion rate per update    | 0.05    |
| Site mutation rate per update    | 0.001   |

## Results

Figure 3 shows the mean training fitness and testing accuracy of the dominant (most fit) individuals from 30 different replicates over time. Fitness and accuracy rise to approximately 96% and 79% over 250,000 updates, respectively. The single best individual accuracy (across the 30 replicate runs) is 81%. High fitness is indicative of high accuracy, although the fitnesses of the individuals are much closer to maximum than their accuracies. The two measures are also strongly correlated. Figure 4(a) demonstrates this in a scatter plot of the relationship between training fitness and training accuracy over time ( $\rho = 0.9255$ ). The correlation is not perfect, however, because our fitness function did not reward



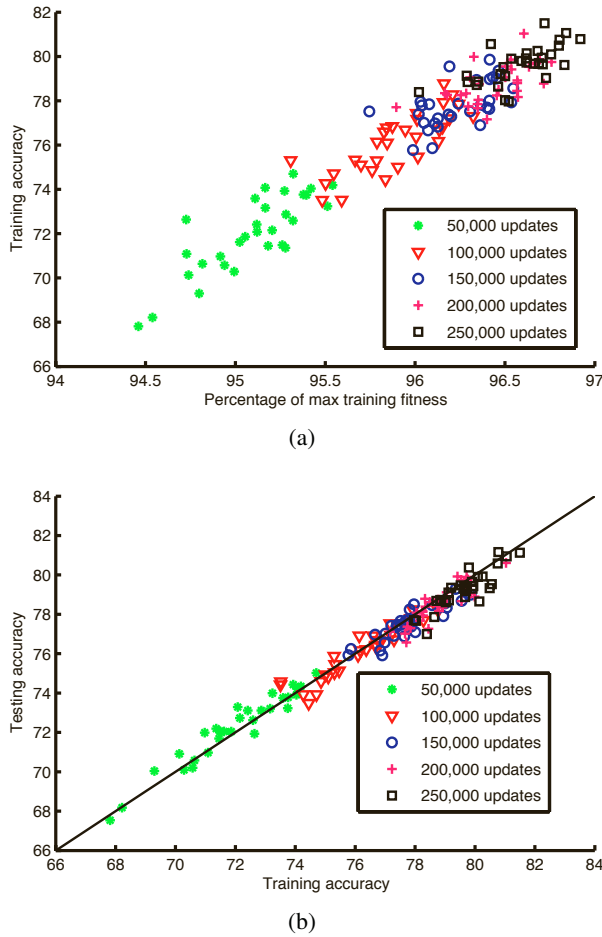


Figure 4: (a): Relationship between training fitness and training accuracy of the 30 replicates over time ( $\rho = 0.9255$ ). Note that training fitness reaches much closer to its maximum than training accuracy. (b): Relationship between training accuracy and testing accuracy of the 30 replicates over time ( $\rho = 0.9757$ ). The  $y=x$  line is shown for clarity, and shows how most of the time training accuracy is higher than testing accuracy, although there are many cases where the reverse is true.

accuracy *per se*, but rather the true positive and true negative rates of the Markov network class decisions. Similarly, as the data in Figure 4(b) shows, accuracy on the training set correlates well with accuracy on the testing set ( $\rho = 0.9757$ ). While training accuracy is often higher than testing accuracy, it can also be lower, and the two accuracies are never different by more than a slight amount. Using the same reasoning as with Figure 4(a), one can say that the discrepancy between training and testing accuracies is due to the fact that the networks were not specifically evolved using accuracy as a fitness function. The results also demonstrate that the networks were capable of generalizing features learned from the training set to the images of the testing set.

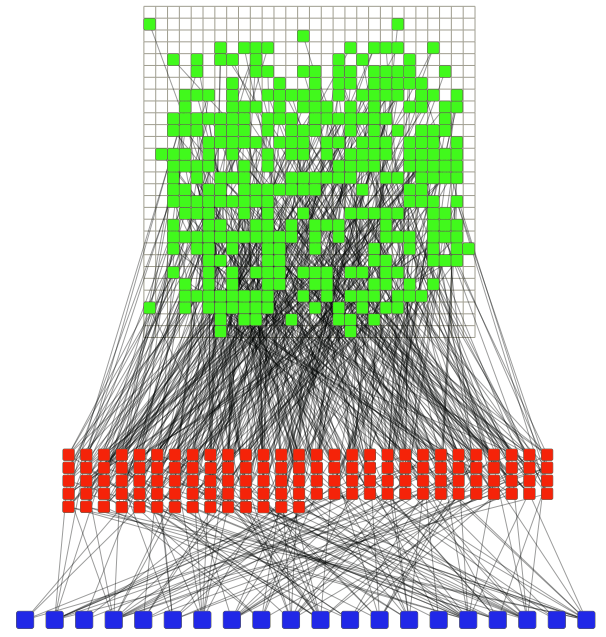


Figure 5: The structure of the Markov network from 250,000 updates with the highest individual testing accuracy. The green squares are the inputs corresponding to the pixels of the 28x28 image field. The red squares are the gates. The blue squares are the output nodes, ordered from  $\{b_{0,0}, b_{1,0} \dots b_{0,9}, b_{1,9}\}$ .

An example evolved Markov network is depicted in Figure 5, with the inputs shown according to their location on the 28x28 image field. The network shown is the individual at 250,000 updates that has the highest testing accuracy, (81.16%). Notably, the network is able to achieve a high relative testing accuracy with a sparse number of inputs. Because of the diversity of the evolutionary process, the structures of the Markov networks that evolved were not all the same as in Figure 5. However, common features of the networks tended to appear. Figure 6(b-d) shows the probabilities of pixels from the 28x28 image field being used as inputs by the 30 dominant individual networks from the replicates at 25,000, 100,000, and 250,000 updates, respectively. The colors range from dark blue to dark red, with a dark blue pixel meaning that no networks had an input at that location, and a dark red pixel meaning that all 30 networks had an input there. It is clear from this progression that certain areas of the image field are favored by the networks more than others. This demonstrates how the Markov networks improve their fitness over time by focusing on certain areas of the image field, and clearly shows the sparsity of the inputs to the networks.

In addition, the entropy of the pixels of the 60,000 training images is shown in Figure 6(a), with dark blue representing zero bits of entropy and dark red representing the max of

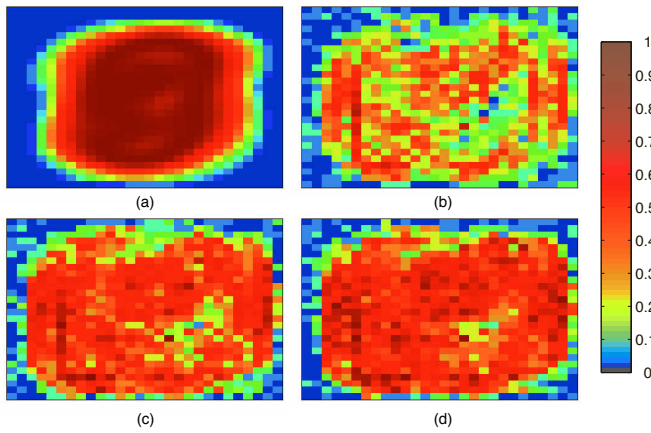


Figure 6: (a): Training image entropy per pixel Eq. (2). Dark blue represents zero bits of entropy and dark red represents the maximum entropy of 1 bit (see colorbar) (b-c): Network input probabilities at 25,000, 100,000, and 250,000 updates, respectively. Colors follow the colorbar.

1 bit. Entropy was calculated in order to compare it to the input probabilities of the networks. Entropy of each of the pixels was calculated using the probability  $p_i$  that a pixel  $i$  was a binary 1 (“turned on”) over all images:

$$H(x) = - \sum_{i=1}^{784} p_i * \log_2(p_i), \quad (2)$$

where  $\text{pixel}_i$  is the probability that a pixel in the set of training images is represented. As shown in Equation 2, entropy of a pixel is higher the closer its probability of being turned on is to 50%. If a pixel was turned on either none or all of the time, its entropy was zero. It was hypothesized that pixels with higher entropy would be more informative about the data set in general, and thus the networks would preferentially connect to these high-entropy pixels. When comparing Figure 6(a) to Figure 6(b-d), this indeed appears to be the case, especially on the border areas.

While the Markov networks focus on certain areas of the image field for input, Figure 6(b-d) also shows how the mean number of inputs of the networks increases over time. Figure 7 depicts a curve showing the mean number of inputs of the networks over time, along with the mean number of gates. Both the mean number of inputs and number of gates increases over time, although after approximately 75,000 updates the increase of both values occurs at a lower rate. Note that because gates can share inputs, there is not a simple linear relationship between the number of gates and the number of inputs. The mean number of outputs per gate also tends to increase, as depicted in Figure 8 (the number of inputs per gate is almost always 4 and is not shown here). It is not a requirement of Markov networks for there to be this concomitant increase in the number of outputs per gate alongside an

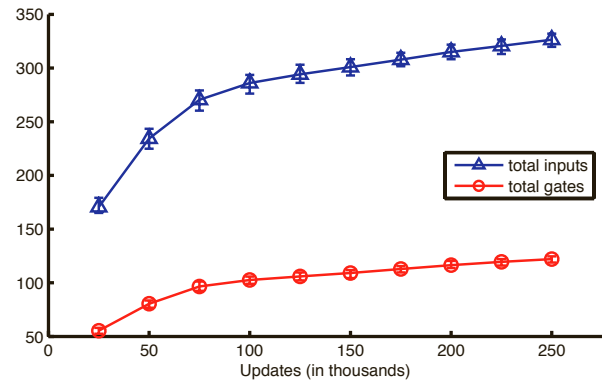


Figure 7: Mean number of inputs (neurons that connect to image pixels) averaged over 30 independent runs (blue, triangles) and mean number of logic gates (red, circles), as a function of evolutionary time. Error bars are constructed using 200-fold bootstrapping with 95% confidence intervals.

increase in the number of inputs and gates. Therefore, this phenomenon in the evolution of the networks suggests that one of the ways that the networks get better at recognizing the images is by having the gates increase the number of outputs that they cover as opposed to focusing on only one output.

Following the evolution of individual Markov networks, we next constructed committees from the dominant individuals from all 30 replicates. A committee is a method for combining the answers of multiple individuals to improve performance, a technique that has been shown to be effective on the MNIST data set (Ciresan et al., 2012). The committee decisions were formed by summing the decisions from each individual. Interestingly, committee results were strongest when individual decisions were allowed to contain votes for multiple classes, i.e., individual votes were allowed to contain ties. At the committee level, the single classification

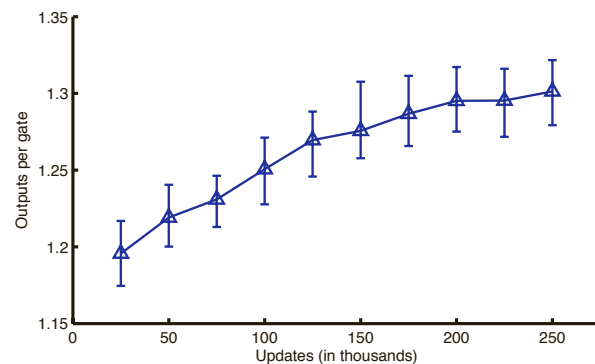


Figure 8: Mean number of outgoing edges per logic gate as a function of evolutionary time. Error bars are constructed using 200-fold bootstrapping with 95% confidence intervals.

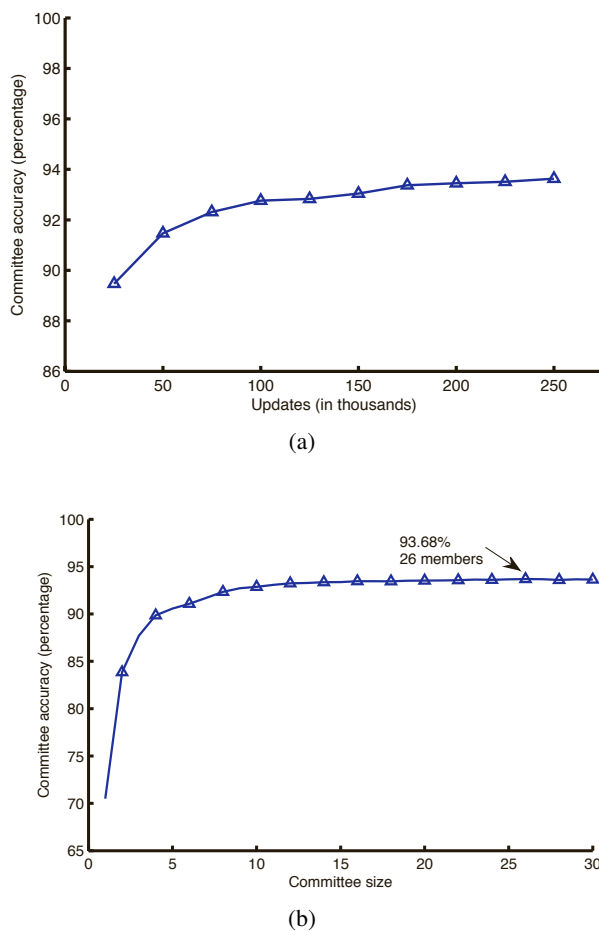


Figure 9: (a): Accuracy of a 30-member committee over evolutionary time. (b): Committee accuracy as a function of committee size, showing diminishing returns. The arrow indicates the most accurate committee.

with the most votes was selected, and ties were again broken randomly. Figure 9(a) shows the 30-member committee accuracy over time on the testing set. The accuracy gradually increases, reaching 93.63% after 250,000 updates.

To examine the effects of committee size on accuracy on the test set, the most-fit individuals from each of the 30 replicates at 250,000 updates were ranked according to how well they did on the training set in terms of fitness, and the highest-ranked  $n$  members assigned to a committee of size  $n$ . We can then determine the dependence of committee accuracy as a function of committee size. Figure 9(b) shows that committee accuracy dramatically increases as committee members are added. We note that a committee of 30 members does not provide the absolute best accuracy for this update (93.63%), although it is very close to the maximum committee accuracy achieved (93.68% with 26 members). These results clearly show the utility of increasing the size of a committee. Although the benefit from adding

new committee members faces diminishing returns, it is remarkable how only a few committee members can dramatically improve accuracy. This suggests that the diversity of the evolutionary process can produce Markov networks that are able to recognize different aspects of the OCR problem, such that pooling their answers in a committee plays on their strengths.

## Conclusion

We have demonstrated an evolutionary approach to image classification that is capable of high accuracy on the MNIST benchmark, while having a small computational footprint. Because the network is essentially a digital circuit, it is easily transferred from one computational environment to another, for example, we have used it on a standard tablet computer to recognize digits drawn by the user on the screen. While the accuracy is probably sufficient for usage in mobile agent navigation, we believe that the image recognition accuracy can be improved significantly. For example, we have only explored a limited set of parameters and fitness functions, and a number of unexplored avenues to improve accuracy remain. For example, there undoubtedly exists a fitness function that shows a better correlation with accuracy than the one used here. Also, pre-processing images to extract salient features (such as lines and corners) could reduce the dependence of class assignments on the locations of specific pixels. Increasing the number of images in the training set by modification of the current set could also help. Indeed, the work by Ciresan et al. (2012) utilized both of these methods.

Perhaps the most promising path to more accurate and invariant image recognition, however, is to create an evolutionary landscape and framework where a more hierarchical image processing algorithm can evolve. In this work, we have limited the processing time (from reading the input image to writing the image classification into the outputs) to exactly one time step. As a consequence, the resulting algorithm cannot possibly use any intermediate Markov variables to build up representations Marstaller et al. (2013) of the concepts. In contrast, a complex visual cortex is hierarchical and processes the simple elements of the image into more and more complex “concepts”, which are then finally used to categorize. We have seen in preliminary work that such a hierarchical organization can emerge, but at the price of a significant slowdown in evolution.

## Acknowledgements

We thank Randal Olson, Fred Dyer, and Rob Pennock for discussions. This work was supported in part by the Paul G. Allen Family Foundation and the National Science Foundation BEACON Center for the Study of Evolution in Action under Cooperative Agreement DBI-0939454. We wish to acknowledge the support of the Michigan State University

High Performance Computing Center and the Institute for Cyber Enabled Research (iCER).

## References

- Bäck, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Oxford, UK.
- Beer, R. (1996). Toward the evolution of dynamical neural networks for minimally cognitive behavior. In P. Maes et al., editor, *Proc. 4th International Conference on Simulation of Adaptive Behavior*, pages 421–429, Cambridge, MA. MIT Press.
- Beer, R. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11:209–243.
- Chau, S., Lafon, S., Shao, J., Szybalski, A. T., and Vincent, L. (2013). Panoramic images within driving directions. U.S. Patent 8,428,873.
- Ciresan, D., Meier, U., Gambardella, L., and Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22:3207–3220.
- Ciresan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3642–3649. IEEE Press.
- Decoste, D. and Schölkopf, B. (2002). Training invariant support vector machines. *Machine Learning*, 46:161–190.
- Edlund, J., Chaumont, N., Hintze, A., Koch, C., Tononi, G., and Adami, C. (2011). Integrated information increases with fitness in the evolution of animats. *PLoS Computational Biology*, 7(10):e1002236.
- Floreano, D. and Keller, L. (2010). Evolution of adaptive behaviour in robots by means of Darwinian selection. *PLoS Biol*, 8(1):e1000292.
- Floreano, D. and Mondada, F. (1998). Evolutionary neuro-controllers for autonomous mobile robots. *Neural Networks*, 11:1461–1478.
- Floreano, D., Suzuki, M., and Mattiussi, D. (2005). Active vision and receptive field development in evolutionary robots. *Evol Comput*, 13(4):527–44.
- Ibañez-Guzmán, J., Laugier, C., Yoder, J.-D., and Thrun, S. (2012). Autonomous driving: Context and state-of-the-art. In Eskandarian, A., editor, *Handbook of Intelligent Systems*, chapter 50, pages 1273–1310. Springer-Verlag London Ltd.
- Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT Press, Cambridge, MA.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324.
- Marstaller, L., Hintze, A., and Adami, C. (2013). The evolution of representation in simple cognitive networks. *Neural Computation*, 25:to appear.
- Nelson, A. L., Barlow, G. J., and Doitsidis, L. (2009). Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57:345–370.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA.
- Pfeifer, R. and Bongard, J. (2006). *How The Body Shapes The Way We Think*. MIT Press, Cambridge, MA.
- Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nat Neurosci*, 2(11):1019–25.
- Salakhutdinov, R. and Hinton, G. (2007). Learning a non-linear embedding by preserving class neighbourhood structure. In *AI and Statistics*, volume 3, page 5.
- Serre, T., Oliva, A., and Poggio, T. (2007). A feedforward architecture accounts for rapid categorization. *Proc Natl Acad Sci U S A*, 104(15):6424–9.
- Stomeo, E., Kalganova, T., Lambert, C., Lipnitsakya, N., and Yatskevich, Y. (2005). On evolution of relatively large combinational logic circuits. In *Proc. NASA/DoD Conference on Evolvable Hardware*, pages 59 – 66.
- Thorpe, S., Fize, D., and Marlot, C. (1996). Speed of processing in the human visual system. *Nature*, 381(6582):520–2.
- Torresen, J. (2001). Two-step incremental evolution of a prosthetic hand controller based on digital logic gates. *Lecture Notes in Computer Science*, 2210:1–13.
- van Dartel, M., Sprinkhuizen-Kuyper, I., Postma, E., and van den Herik, J. (2005). Reactive agents and perceptual ambiguity. *Adaptive Behavior*, 13:227–42.
- Vassilev, V. and Miller, J. (2000). Scalability problems of digital circuit evolution evolvability and efficient designs. In *Proceedings of the Second NASA/DoD Workshop on Evolvable Hardware*, pages 55–64. IEEE.



# Multiple Time Scales Observed in Spontaneously Evolved Neurons on High-density CMOS Electrode Array

Eiko Matsuda<sup>1,7</sup>, Takeshi Mita<sup>2</sup>, Julien Hubert<sup>1</sup>, Mizuki Oka<sup>3</sup>,  
Douglas Bakkum<sup>4</sup>, Urs Frey<sup>5</sup>, Hirokazu Takahashi<sup>6</sup> and Takashi Ikegami<sup>1</sup>

<sup>1</sup>Department of Arts and Sciences, the University of Tokyo, Tokyo, Japan, <sup>2</sup>Department of Mechano-informatics, the University of Tokyo, Tokyo, Japan,

<sup>3</sup>Tsukuba University, Ibaraki, Japan, <sup>4</sup>ETH Zurich, Department of Biosystems Science and Engineering, Basel, Switzerland,

<sup>5</sup>RIKEN Quantitative Biology Center, Kobe, Japan, <sup>6</sup>Research Center for Advanced Science and Technology, the University of Tokyo, Tokyo, Japan,

<sup>7</sup>JSPS research fellow, email: eiko@sacral.c.u-tokyo.ac.jp

## Abstract

Spontaneous evolution of neural cells was recorded around 4-34 days in vitro (DIV) with high-density CMOS micro-electrode array, which enables detailed study of the spatio-temporal activity of cultured neurons. We used the CMOS array to characterize 1) the evolution of activation patterns of each putative neurons, 2) the developmental change in cell-cell interactions, and finally, 3) emergence of multiple timescales for neurons to exchange information with each other. The results revealed not only the topology of the physical connectivity of the neurons but also the functional connectivity of the neurons within different time scales. We finally argued the relationship of the results with “functional networks”, which interact with each other to support multiple cognitive functions in the mature human brain.

## Introduction

How can the gap between living and nonliving matter be bridged? Since 1987 when Artificial Life was launched by Christopher Langton, we have not answered this question. Rodney Brooks wrote in the paper (2001) that there are four possibilities why we still cannot make living machines: 1) An alife model is correct, but several parameters were set incorrectly; (2) an alife model needs more complexity; (3) we need more computational power; and (4) a new fundamental law in addition to the laws of physics is needed. In this paper, we search for the possibility of (2) yet unrevealed laws of neuro-dynamics, by studying cultivated neural cells on a glass plate.

Biological neurons are cultivated on a glass plate from neural “seeds”. The seeds develop into either neural or glia cells. Neurons have cell bodies, axons, and numerous dendrites, which the neurons use to connect with each other. A unique characteristic of the present study is that we record the action potential from neurons by using a CMOS array glass plate. As we will describe later, each CMOS is the same size order of the neurons. Therefore, by using the CMOS array, we can potentially accurately record the time series of each neural firing. A remarkable aspect of this biological neural network is the developmental process. The entire time course of the growing process can be recorded

with the CMOS array. We analyze the time series data to characterize the developmental dynamics.

A disadvantage of this experiment is that we have no way of designating which neurons connect to which. We thus measure the information transfer from the time series to infer the neural connectivity. This method reveals not only the topology of the physical connectivity of the neurons but also the functional connectivity of the neurons within different time scales. A finding in this paper is that growing biological neurons use different time scales to exchange information with each other.

The paper is organized as follows. In section of Materials and Methods, the specifications of the CMOS array and the biological conditions for the neural cells are provided. The method for cultivating cells and associated techniques are also described. In section of Result and Discussion, the analyzed results are presented. The activation patterns of each cell are quantified with inter-spike intervals (ISIs). Cell-cell interaction is also inferred by transfer entropy (TE), which reveals that multiple functional networks emerge from the neural original network. Finally, in section of Conclusion, the paper is summarized, and future work is discussed.

## Materials and Methods

To measure the electrical activity of cultured neurons, we used a high-density CMOS microelectrode array <sup>1</sup>(Frey et al., 2010). The CMOS array is pictured in figure 1 (a). This array is an emerging instrument for investigating the spatio-temporal activity of cultured neurons in detail. The CMOS array has 11,011 recording sites with an inter-electrode distance of 18  $\mu\text{m}$ , i.e., in the order of cell body size, and a sampling rate of 20 kHz. This high spatio-temporal resolution allows precise recording of action potentials from the identified cell bodies of neurons. Using this high spatial resolution, we localized neural somata and recorded their activity.

<sup>1</sup>All procedures were approved by the institutional committee at the University of Tokyo, and were performed in accordance with the Guiding Principles for the Care and Use of Animals in the Field of Physiological Science of the Japanese Physiological Society.

This CMOS array is superior to conventional microelectrode array (MEA) (Sun et al., 2010; Eytan and Marom, 2006; van Pelt et al., 2004) in respect of spatio-temporal resolution: The locations of recording sites in conventional MEAs are predetermined, with an inter-electrode distance of  $200\text{ }\mu\text{m}$ , so that it is difficult to identify signals from an individual cell, and neurons far from these recording sites are not included. Alternatively, optical imaging can be used to study the  $\text{Ca}^{++}$  dynamics of any neuron of interest; however, the temporal resolution is not high enough to characterize the action potentials of each neuron.

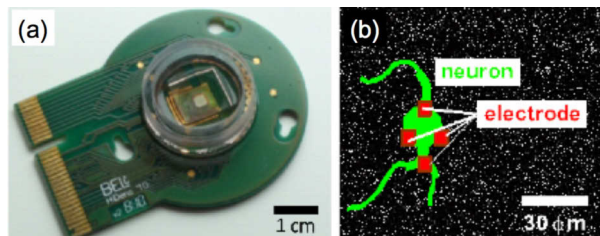


Figure 1: High-density CMOS array (a) Appearance. (b) Interval of each electrodes is close to the cell body size.

**Dissociated neural culture** The neural cultures were prepared from the cerebral cortex of E18 (embryonic 18) Wistar rats. The cortex was triturated with trypsin and dissociated cells were plated and cultured on high-density CMOS microelectrode arrays coated with polyethylenimine and laminin. For the first 24 h, the cells were cultured in neurobasal medium containing 10% horse serum, 0.5 mM GlutaMAX and 2% B-27 supplement. After the first 24 h, half of the medium was replaced with growth medium in the form of Dulbecco's modified Eagle's medium with 10% horse serum, 0.5 mM GlutaMAX, and 1 mM sodium pyruvate. During the cell culturing, half of the medium was replaced once a week with the growth medium. The cultures were placed in an incubator at  $37\text{ }^{\circ}\text{C}$  with an  $\text{H}_2\text{O}$ -saturated atmosphere consisting of 95% air and 5%  $\text{CO}_2$ . We prepared for two conditions with different densities of neurons. The denser chips had 35,000 cells plated, denoted as Chip#1D and Chip#2D, while the sparser ones had 14,000 cells, termed Chip#1S and Chip#2S.

**Recording of neural activity** Neural activity was recorded with high-density CMOS microelectrode arrays. Before the neural somata activity was recorded, almost all of the 11,011 electrodes were scanned to obtain an electrical activity map with which we estimated the locations of the somata. A scan session consisted of 95 recordings; each recording was conducted for 60 sec with about 110 electrodes that were selected randomly, while avoiding overlap with already selected electrodes. An

electrical activity map was obtained from the scanned data by calculating the average height of the spikes for every electrode. We assumed that the neural somata were near the local peaks in the Gaussian-filtered electrical activity map. About 100 of the higher-level peaks were selected, and then the nearest electrodes were selected for recording. If the number of local peaks was fewer than 126, then all the peaks were selected. The electrical activity of the selected electrodes was recorded for 30 min. All recordings were done at a 20-kHz sampling rate using the LimAda spike detection algorithm (Wagenaar et al., 2005) with a threshold of 5. Unexpected double-detected spikes were removed from the data before the analysis was conducted.

## Results and Discussion

### Simple observation of activity patterns

First, we observed the activation patterns of the neurons by examining the time series of the neural spikes. Figure 2 shows examples of raster plots of Chip#1D and Chip#1S. Chip#2D and Chip#2S showed similar tendency to their respective counterparts (data not shown). The data plotted here are a compressed version of the raw data with a different bin-length, which is denoted by  $\Delta t$ . Namely, the spikes within the same bin are regarded as one spike. If  $\Delta t$  is smaller, the time series represents single spikes generated from each single neuron, while a larger  $\Delta t$  represents the macroscopic behavior created by an ensemble of neurons. In the figure 2, the data are plotted in  $\Delta t = 0.6\text{ ms}$  and  $100\text{ ms}$ . At the top of each raster plot, the activation ratio is displayed. We chose  $\Delta t = 0.6\text{ ms}$  and  $100\text{ ms}$  as the examples by two reasons; a single spike of neurons lasts around  $1\text{ ms}$ , while, observably from figure 2, synchronous activation of neurons is detected around  $\Delta t = 10\text{--}100\text{ ms}$ . Therefore, we took  $\Delta t = 0.6\text{ ms}$  to capture interaction between individual cells (microscopic), while  $\Delta t = 100\text{ ms}$  is to observe collective activities generated from an ensemble of neurons (macroscopic).

As shown in figure 2(a) right, the Chip#1D neurons at days *in vitro* 7 (DIV 7) are activated intermittently in synchronization. This activation pattern was burst synchronization, in which an ensemble of neurons has active and silent phases; in the active phase, synchronous activation of the neurons is intermittently observed. The burst synchronization is a typical activation pattern observed in cell cultures (Maeda et al., 1995). On the left of figure 2(a), the silent phase of the neural activation is magnified with  $\Delta t = 0.6\text{ ms}$ . Neural spikes are observed sparsely, which suggests that each neurons was activated independently from others.

On the other hand, at the later stage (DIV 14) of the Chip#1D, burst synchronization is not observed (figure 2 (b) right). Instead, we observed each neurons shows different activation patterns; i.e., some neurons were activated

almost all the time, some showed spikes less often, and others remained silent. These neurons seem to be activated at different frequencies. The left of figure 2(b) shows that, at  $\Delta t = 0.6$  ms, a single spike is followed by another, which suggests that one single spike can activate another. In contrast, Chip#1S showed the burst synchronization throughout the entire recoding period (figure 2(c) and (d)).

To summarize the results thus far, burst synchronization was observed in two conditions; in the earlier developmental stage of the dense cell condition, or in the sparse cell condition. In those cases, any single spikes hardly activate another. In the later stage of the dense cell condition, the neurons spiked at various frequencies, where single spikes induce others. This tendency is explained by the maturity of synapses. At the earlier developmental stage, the synapses are not mature; therefore, a single spike cannot activate another. Still, if the neurons send spikes at the same time (synchronous bursts), the signals become enough strong to activate each other. In contrast, at the later stage of development, the synapses were enough strong to transmit a single spike from one neuron to another.

### Single cell activity evaluated with ISI

In the next step of the analysis, we quantified the activation patterns of each neuron. Therefore, we analyzed the interspike intervals (ISI) distribution of the neural activity. Figure 3 (a) depicts examples of ISI distribution recorded from putative single neurons cultured in Chip#1D. In the earlier culture stage (DIV 7; figure 3 (a) left), exponential decay was observed, while the neurons tended to obey the power law at DIV 14 (right figure). To quantify this tendency, we plotted the ISI frequency on a logarithmic scale, and we fit the slope with a straight line by the least squared method. If the ISI distribution follows the power law, the distribution should be a straight line, so that the R-squared value is an index of fitting the power law.

Figure 3(b) shows the average of the R-squared values over each neuron; the values tended to increase in Chip#1D, Chip#1S and Chip#2S. Especially, Chip#1D shows a drastic increase at DIV 10, which suggests the neural activity became closer to the power law. Chip#2D also shows a higher R-squared value throughout the entire recording period. The recordings for Chip#2D started from DIV 17, which explains the result is similar to the later stage of Chip#1D.

The power law means that each neuron exhibited a wide range of ISIs. It may be related to the observation from the raster plots (figure 2), which showed that a wide range of frequencies between neurons were observed at the later development stage. To sum them up, a broader range of time scales likely emerges after synaptic maturation. However, from this ISI analysis, it is not possible to understand how those cultured neurons interact with each other to generate the activity patterns. In the next part, we then investigated neuron connectivity.

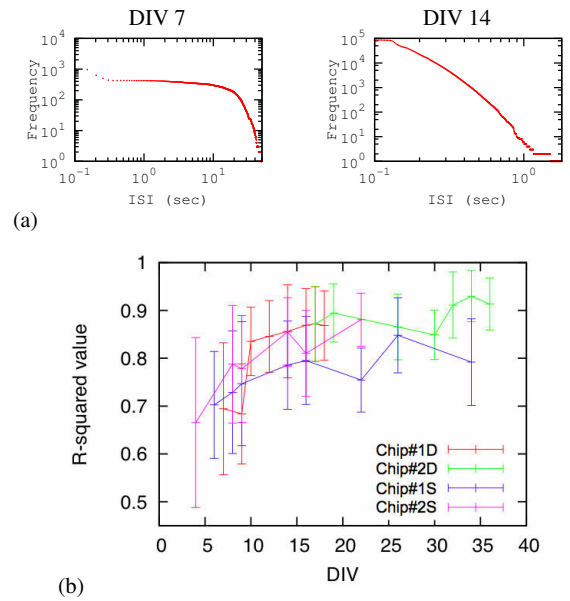


Figure 3: (a) Examples of the ISI distribution of neural activity recorded in a cell on Chip#1D. The X and Y axes depict the ISI logarithm and frequency respectively. The two figures show the results for each 7 and 14 DIV. The estimated exponents for DIV 7 and 14 are -1.40(0.57) and -3.77(0.57). The values inside the parentheses denote the R-squared values of the regression lines. (b) Change in the R-squared values with DIV. The R-squared value is obtained from a regression line fit to the ISI distribution, which is plotted on a logarithmic scale. When the R-squared value is closer to 1.0, the ISI distribution is more likely to obey the power law.

### Cell-cell interaction inferred with transfer entropy

We used transfer entropy (TE) to estimate the effective connectivity for transferring information from one neuron to another. TE measures directed information transfer, which detects causal relationship between two time series (Schreiber, 2000; Lizier et al., 2011; Staniek and Lehnertz, 2008; Bertschinger et al., 2008). For instance, higher TE from one neuron to another indicates that the first neuron strongly affects the second. Therefore, TE enables us to find the functional synaptic connectivity. We defined the TE, and then applied it to artificial neural networks to ensure the validity of TE to estimate effective connectivity. Finally, we applied TE to the cultured neural cells to infer their topology.

**Definition of transfer entropy** Information is measured with Shannon entropy, which quantifies the amount of uncertainty associated with a system  $X$ . Specifically, Shannon entropy of a system  $X$  is defined as:

$$H(X) = - \sum_{x \in X} p(x) \log p(x), \quad (1)$$

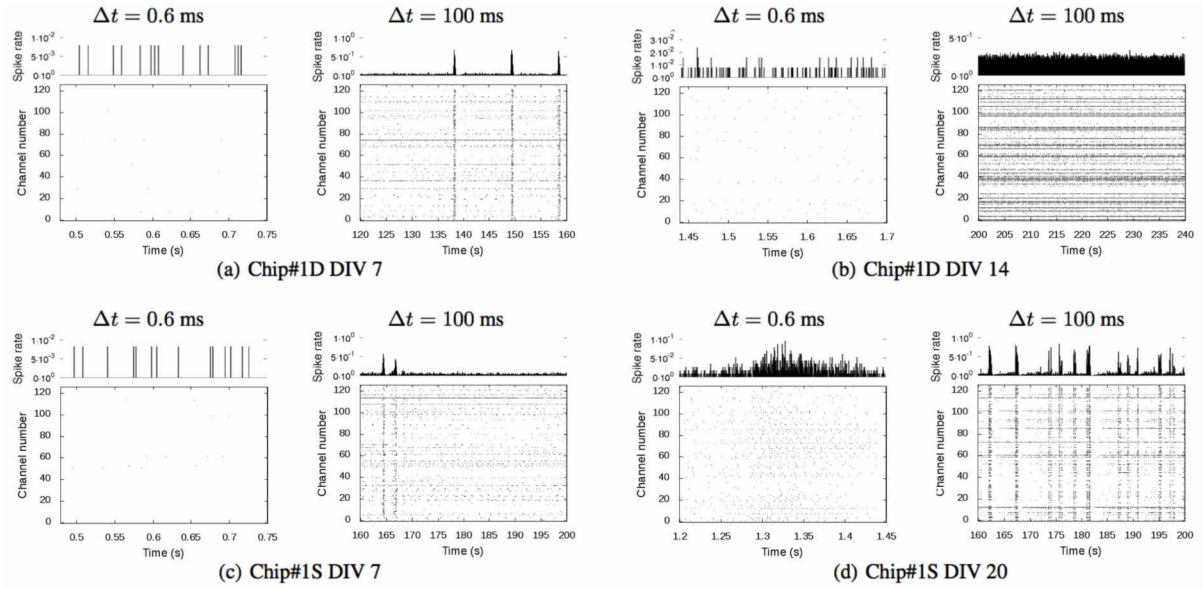


Figure 2: Examples of the raster plot of the cultured neurons. The X axis denotes time (s). The Y axis represents the indexes of the recoding channel, where one channel can be considered as one neural cell. The subfigures at the top of the raster plots shows the spike rate. (a) - (d) Results for Chip#1D and Chip#1S with different DIVs.  $\Delta t = 0.6$  ms (microscopic) and 100 ms (macroscopic).

where  $p(x)$  denotes the probability of  $x$  ( $x$  is an event of  $X$ ). To evaluate the dependency between  $X$  and  $Y$ , mutual information ( $MI$ ) is defined as follows:

$$\begin{aligned} MI(X, Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X). \end{aligned} \quad (2)$$

$H(X|Y)$  means conditional entropy, i.e., the uncertainty of  $X$  when  $Y$  is known. Therefore,  $MI(X, Y)$  suggests a decrease in the uncertainty of  $X$  when  $Y$  is known.  $MI(X, Y)$  measures the dependency between  $X$  and  $Y$ ; so that this variable cannot quantify a causal relationship between them.

TE measures the causal relationship between  $X$  and  $Y$  by calculating the past history. The TE from  $X$  to  $Y$  is denoted by  $T_{X \rightarrow Y}$ , which is written as:

$$\begin{aligned} T_{X \rightarrow Y} &= H(y_{n+1}|y_n^{(k)}) - H(y_{n+1}|y_n^{(k)}, x_n^{(l)}) \\ &= \sum_{n=0}^{N-1} p(y_{n+1}, y_n^{(k)}, x_n^{(l)}) \log \frac{p(y_{n+1}|y_n^{(k)}, x_n^{(l)})}{p(y_{n+1}|y_n^{(k)})} \end{aligned} \quad (3)$$

where  $n$  is the current time step, and  $y_t^{(k)}$  and  $x_t^{(l)}$  are the past variables with length  $k$  and  $l$  respectively (i.e.,  $y_t^{(k)} = \{y_t, y_{t-1}, \dots, y_{t-k+1}\}$  and  $x_t^{(l)} = \{x_t, x_{t-1}, \dots, x_{t-l+1}\}$ ). When the next step of  $Y$  ( $= y_{n+1}$ ) is conditioned from the past history of  $X$  ( $= x_n^{(l)}$ ), then  $H(y_{n+1}|y_n^{(k)}, x_n^{(l)})$  takes a smaller value than  $H(y_{n+1}|y_n^{(k)})$ . If  $y_{n+1}$  is independently determined from

the past history of  $X$ , then the two components will have the same value. Therefore,  $TE_{X \rightarrow Y}$  measures the causality of  $X$  to  $Y$ .

**Settings of artificial neural networks** TE analysis was first applied to a computational neural model. The model was built around Izhikevich neurons connected through artificial synapses (Izhikevich, 2003). The Izhikevich neurons form a simple model of cortical neurons that is implemented by a system of two differential equations modeling the membrane potential and the refractory period. When the membrane potential reaches a threshold value (for instance, 30 mV), a spike is emitted. This spike is transferred to post-synaptic neurons through some shared synapses. The voltage on arrival is the original spike strength, modulated by the efficacy of the synapses. For instance, an initial spike of 30 mV traveling on a synapse with an efficacy of 0.5 delivers a voltage of 15 mV to the post-synaptic neuron. Every synapse has a delay of 1 ms between the time of emission and the arrival of a spike.

The complete model is composed of seven neurons: four input neurons receiving randomly generated external stimulations, two internal neurons and one output neuron. The parameters for the Izhikevich neurons correspond to the regular spiking model ( $a = 0.02$ ,  $b = 0.2$ ,  $c = -65$  mV and  $d = 6$ ). Different types of connectivity patterns have been tested, ranging from fully inter-



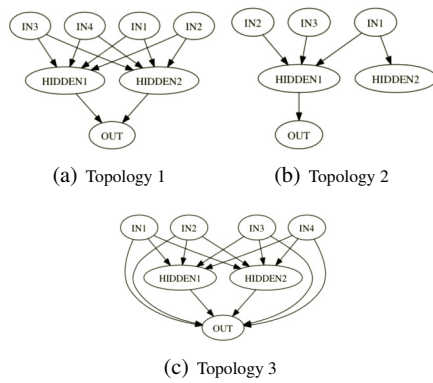


Figure 4: Topology of the original network used in the computer simulation.

connected to sparse (figure 4 (a)–(c)). The strength of the connection is randomly assigned based on uniform distribution. Every update of the model represents a 0.1-ms step in time, which ensures the model’s mathematical stability. The total duration of a test is 1000 s, which corresponds to 10,000,000 updates of the model.

**Estimated connectivity of the artificial network** From the time series of the artificial neural activity, we calculated the TE from one neuron to another. Using the TE, we estimated the network structure of the artificial neurons. A synaptic connection from one to the other was assumed when the TE between two neurons was higher than the threshold. Then, we compared the topology of the reconstructed network and the original network shown in figure 4 (a)–(c). The number of false edges for each topology is shown in figure 5.  $\Delta t$  in this figure is the same one as previously used, which represents the bin-length of the compressed time series. This figure shows that the optimal parameter set to reconstruct the original topology depends on the dimension of the past variables ( $k$  and  $l$ ) and  $\Delta t$ .

The result ensures that the effective connectivity to transmit signals by TE is estimated. However, a good approximation depends on the dimension and time scale  $\Delta t$ . We used various  $\Delta t$  to get a good approximation of the effective connectivity in the cultured neurons.

**Estimated connectivity of cultured neurons** Figure 6 shows some examples of the estimated network structure of neurons. This depicts the network of Chip #1D DIV 14 with different  $\Delta t$  ( $= 0.6$  ms, 1 ms and 10 ms). An edge is drawn if the TE from one neuron to another is higher than the threshold, which was set to 0.00001. The threshold is determined arbitrarily, but to display dynamical change in connectivity patterns. The first observation about this figure is that different topology is structured depending

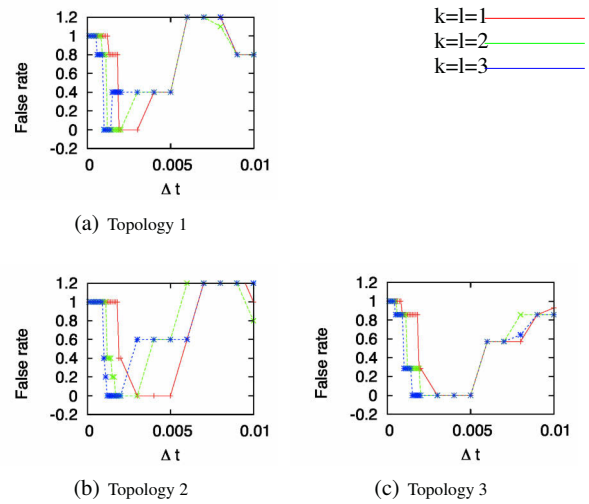


Figure 5: Number of false connections with different  $\Delta t$  values. The threshold was set to 0.005.

on  $\Delta t$ . As is shown with the artificial neural network, each connection may not have the same optimal  $\Delta t$  to estimate connections. Based on (Oka and Ikegami, 2013), we used the optimal  $\Delta t$  to understand the information flow in the network.

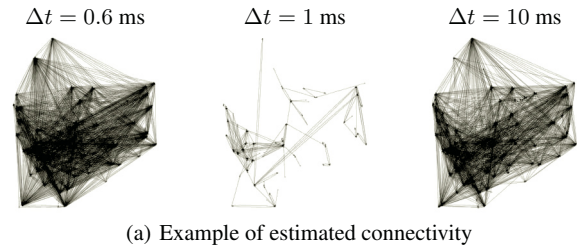


Figure 6: Examples of the network structure of the information flow obtained from TE. The data used here is from Chip#1D DIV 14.  $\Delta t = 0.6$  ms, 1 ms, and 10 ms. The TE threshold to draw edges was set to 0.00001.

### Optimal time scale to convey information

As observed, the cell-cell interaction estimated from the TE analyses depends on the time scale  $\Delta t$ , so that we evaluated the optimal  $\Delta t$  to estimate the network structure better. We defined the optimal  $\Delta t$  as  $\Delta t^*$  that maximizes TE.  $\Delta t^*$  was calculated for each directed pair of neurons. Information was considered to be transferred most effectively with the  $\Delta t^*$ . Therefore, the  $\Delta t^*$  exhibits effective synaptic connectivity from each neuron to another.

Figure 7 (a) and (b) depict  $\Delta t^*$  from one neuron to another in Chip#1D and Chip#1S respectively. Edges are

drawn if TE at the  $\Delta t^*$  is larger than the threshold ( $= 0.00001$ ; same as the one used above). The color indicates the value of  $\Delta t^*$ , where red represents  $\Delta t^*$  smaller than 10 ms and blue indicates larger. We defined  $\Delta t^* = 10$  ms as a threshold to assign color, because burst activity was observably detected with  $\Delta t$  larger than 10 ms (figure 2). Hereafter, we call  $\Delta t > 10$  ms “the smaller  $\Delta t^*$ ”, and  $\Delta t \geq 10$  ms “the larger  $\Delta t^*$ ”, where we divide the activity pattern into microscopic and macroscopic using this threshold. Additionally, figure 8 shows the  $\Delta t^*$  distribution.

Figure 7 (a) and 8 (a) show that, at an earlier stage of the development of Chip #1D, a larger  $\Delta t^*$  is mainly observed. The number of the smaller  $\Delta t^*$  increases at DIV 12, which means a causal relationship between each neurons is observed within a shorter period. This is consistent with the observation in figure 2 (b) that a single spike activates another neuron within a short period. For Chip#2D, a smaller  $\Delta t^*$  was observed throughout the entire recording period (figure 8(b)). Chip#2D was recorded beginning at DIV 17, which is relatively later compared with those of the other chips. That can explain why Chip#2D was similar to the later stage of Chip#1D.

Interestingly, the smaller  $\Delta t^*$  observed in Chip#1D and Chip#2D coexisted with the larger  $\Delta t^*$  (figure 8 (a) and (b)). This suggests that neural activity occurs at a wider range of time scales. Figure 7 (a) shows that the intervening neurons have red and blue edges, which implies microscopic and macroscopic activity to exchange information with each other. However, the frequency of the smaller  $\Delta t^*$  in Chip#1S and Chip#2S was low throughout the entire recording period (figure 7(b), 8(c-d)). Still, similarly to the result for the dense cell condition, as DIV increased, the number of the smaller  $\Delta t^*$  grew.

## Conclusion

We characterized neural activity cultured on a high-density CMOS array. The neurons showed different activation patterns depending on the density and age (days *in vitro*; DIV). The neurons did not receive external inputs, but still spontaneously evolved. The neurons were cultured in two different conditions, i.e., dense or sparse.

When the cell density was high, the neurons showed burst activity first, and afterward, each neural cell activated at a different frequency (figure 2 (a), (b)). Some were activated all the time, some showed spikes intermittently, and others remained silent. The ISI distribution of each cell became closer to the power law as the DIV increased (figure 3), which suggests a single cell evolved to show a wider range of time scales. These results suggest that neural activity exhibits a wider range of time scales as the synapses mature. This tendency was shown clearly with TE (figure 7). At the earlier stage of development, TE was mainly optimized at a larger time scale, while, at the later stage, TE was also optimized at a smaller time scale (figure 8).

A mature human brain is a collection of functional networks, each of which corresponds to a different cognitive function (Fair et al., 2009). In this paper, we insist that even a neural network on a glass plate spontaneously develops “functional networks”, which can be distinguished in terms of the time scale determined by effective information transfer. Without relevant sensory input, we cannot say the networks are functional in the proper context of brain science; however, we speculate that the spontaneous development of “functional networks” is a candidate for the brain functional network. In future work, we will connect the neurons with a navigation robot to see how the functional networks actually “function” as cognitive modules.

## Acknowledgements

This work was supported by Grant-in-Aid for Scientific Research (Studies on Homeo-Dynamics with Cultivated Neural Circuits and Embodied Artificial Neural Networks; 24300080), Grant-in-Aid for Scientific Research (23680050), and Denso Corp. Eiko Matsuda thanks the Japan Society for the Promotion of Science for research funding.

## References

- Bertschinger, N., Olbrich, E., Ay, N., and Jost, J. (2008). *Biosystems*, 91(2):331–34.
- Eytan, D. and Marom, S. (2006). Dynamics and effective topology underlying synchronization in networks of cortical neurons. *The Journal of Neuroscience*, 26(33):8465–8476.
- Fair, D. A., equal contributor, A. L. C., Power, J. D., Dosenbach, N. U. F., Church, J. A., Miezin, F. M., Schlaggar, B. L., and Petersen, S. E. (2009). Functional brain networks develop from a “local to distributed” organization. *PLOS COMPUTATIONAL BIOLOGY*, 5:e1000381.
- Frey, U., Sedivy, J., Heer, F., Pedron, R., Ballini, M., Mueller, J., Bakkum, D. J., Hafizovic, S., Faraci, F. D., Greve, F., et al. (2010). Switch-matrix-based high-density microelectrode array in cmos technology. *IEEE Journal of Solid-State Circuits*, 45(2):467–482.
- Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572.
- Lizier, J. T., Heinzle, J., Horstmann, A., John-Dylan, Haynes, and Prokopenko, M. (2011). Multivariate information-theoretic measures reveal directed information structure and task relevant changes in fmri connectivity. *Journal of Computational Neuroscience*, 30(1):85–107.
- Maeda, E., Robinson, H. P. C., and Kawana, A. (1995). The mechanisms of generation and propagation of synchronized bursting in developing networks of cortical neurons. *Journal of Neuroscience*, 15(10):6834–6845.
- Oka, M. and Ikegami, T. (2013). Exploring default mode and information flow on the web. *PLOS ONE*, 8(4):e60398.
- Schreiber, T. (2000). Measuring information transfer. *Phys Rev Lett*, 85:464–464.

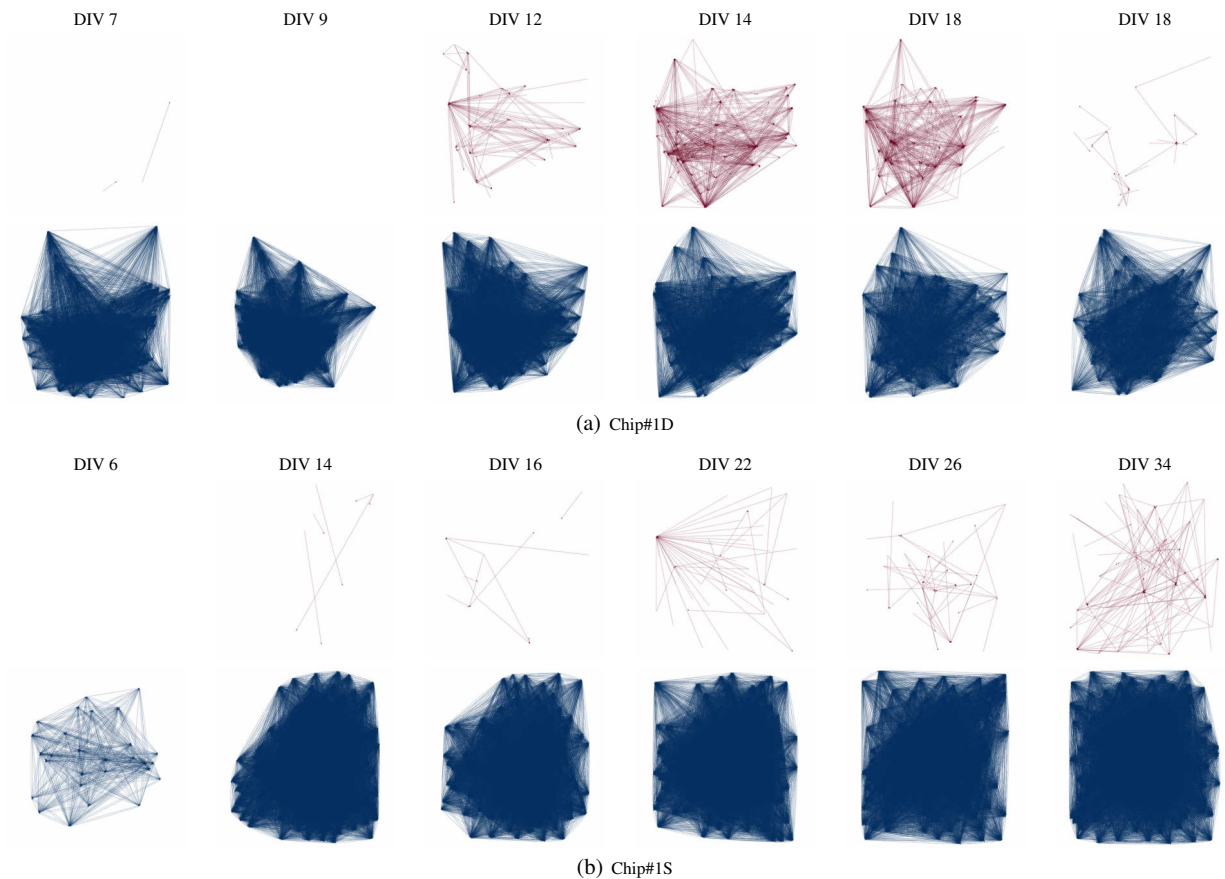


Figure 7: Network structure obtained with  $\Delta t^*$  from one neuron to another. The TE threshold for drawing edges is also equal to 0.00001. The color indicates the value of  $\Delta t^*$ , where red represents a  $\Delta t^*$  smaller than 10 ms, and blue is for a larger  $\Delta t^*$ .

- Staniek, M. and Lehnertz, K. (2008). *Phys. Rev. Lett.*, 100(15):158101.
- Sun, J.-J., Kilb, W., and Luhmann, H. J. (2010). Self-organization of repetitive spike patterns in developing neuronal networks *in vitro*. *European Journal of Neuroscience*, 32:1289–1299.
- van Pelt, J., Wolters, P. S., Corner, M. A., and Rutten, W. L. C. (2004). Long-term characterization of firing dynamics of spontaneous bursts in cultured neural networks. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 51(11):2051–2062.
- Wagenaar, D. A., DeMarse, T. B., and Potter, S. M. S. M. (2005). Meabench: A toolset for multi-electrode data acquisition and on-line analysis. In *Conference Proceedings of 2nd International IEEE EMBS Conference on Neural Engineering*, pages 518–521, Pasadena, CA.

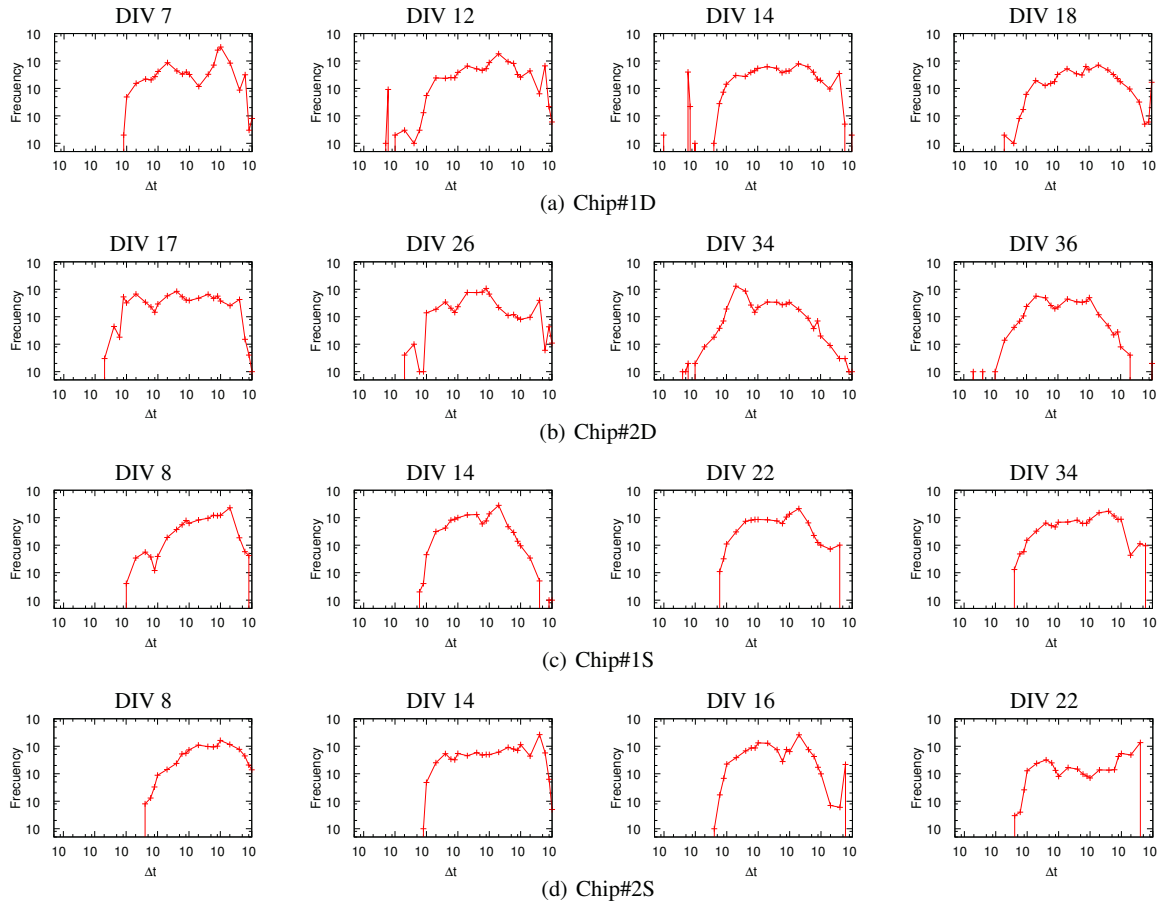


Figure 8: Distribution of  $\Delta t^*$  with different DIVs. The X axis denotes  $\Delta t^*$ , while the Y axis shows the frequency of  $\Delta t^*$  for each connection of neurons. (a) Chip#1D often shows smaller  $\Delta t^*$  after DIV 12, while (b) Chip#2D displays smaller  $\Delta t^*$  throughout the entire recoding periods. (c) (d) Smaller  $\Delta t^*$  is observed less often. Still,  $\Delta t^*$  decreases as DIV increases.



## Sequence selection and evolution in a binary polymer model

Steen Rasmussen<sup>1,2</sup>, Harold Fellermann<sup>1</sup>, Shinpei Tanaka<sup>3</sup>

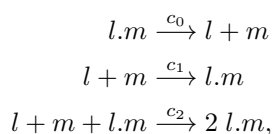
<sup>1</sup>Center for Fundamental Living Technology, University of Southern Denmark, Campusvej 55, 50230 Odense, Denmark

<sup>2</sup>Santa Fe Institute, Santa Fe, 1399 Hyde Park Rd, New Mexico, USA

<sup>3</sup>Graduate School of Integrated Arts and Sciences, Hiroshima University, 1-7-1 Kagamiyama, Higashi-Hiroshima 739-8521, Japan  
 steen@sdu.dk, harold@sdu.dk, shinpei@hiroshima-u.ac.jp

The emergence of autocatalytic structures in model chemistries has been a prominent subject throughout the history of artificial life research (Rasmussen (1985); Farmer et al. (1986); Kauffman (1986); Rasmussen (1989); Hordijk et al. (2011) and others). Most of these works have been concerned with the likelihood of finding autocatalytic sets in a population of random cross-catalytic molecules. Here, in contrast, we study how the detailed sequence structure determines the properties of the emergent cooperative structures. In particular, we study a system of binary polymers, where each polymer can replicate itself by exact ligation of two matching subsequences. We report the emergence of stable cooperative structures with high equilibrated polymer concentrations together with a quantitative connection between the details of the sequence and the frequency as well as the stability of the evolving cooperative structures. These findings could have implications for early earth information polymers as well as the design of protocell information polymer networks.

In the simplest realization, we allow for decomposition, random ligation, and autocatalytic ligation of polymers via the three reactions



where  $l$  and  $m$  are strings of arbitrary length over the alphabet  $\{0, 1\}$  and  $l.m$  denotes string concatenation.  $c_0$ ,  $c_1$ , and  $c_2$  are the respective reaction rates.

If random ligation is comparably rare ( $c_1 \ll c_2 \bar{x}$  for some typical species size  $\bar{x}$ ), the system exposes several unexpected dynamics: (1) out of exponentially many possible strings, stochastic simulation (Gillespie (1977)) repeatedly selects only very few specific strands; (2) the selected populations are strikingly regular — with the motif 010101 being most common; (3) the occupancy of most strings in a population fluctuates around a constant value independent of the length of the string (c.f. Fig. 1) All these properties are in direct contrast to a scenario without catalysis ( $c_2 = 0$ ).

Most of these features can be explained analytically. Examining the reaction kinetic equations reveals that every string of a stationary state must be accompanied by its substrings which are generated through decomposition. We call the longest members of a population its “chiefs” and the sets of their substrings “clans”. In the limit  $c_1 \rightarrow 0$ , chiefs can have an arbitrary occupancy, whereas all non-chief members of a population equilibrate to a constant value determined by the rates of autocatalysis and decomposition as  $\bar{x} = \sqrt{c_0/c_2}$ . Linear stability analysis of several hundred exemplary stationary states confirms that these populations are indeed stable states. Under purely random ligation ( $c_2 = 0$ ), on the other hand, the stationary strand distribution is exponential in strand length  $k$ :  $x_k = (c_0/c_1)e^{-b|k|}$ , where  $b$  is given by the boundary condition.

When starting from a pool of monomers, the ligation dynamics will transport most of the material from the monomers to the emerging front of a chief-clan structure. Rare random ligations slowly expand this front by forming new chiefs. To explain why regular chiefs, such as 01010101, are selected more frequently than irregular ones we note: (i) regular chiefs require the formation of fewer intermediate chiefs: formation of 01010101, for example, requires three random ligations, whereas formation of 01101100 requires at least five ligations; (ii) regular chiefs offer more reaction pathways than irregular ones, as substrings of irregular strings have to ligate in correct order; (iii) population size strongly impacts the reaction rates which results in selection of certain regular sequence patterns.

Thoroughly calculating the likelihoods of forming given chief structures from monomer pools by occasional random ligation and subsequent equilibration, confirms quantitatively that this is indeed the driving mechanism of selection in the simulation results. These calculations also suggest that the “twotowers” structure (Fig. 1.d) becomes less likely with increasing system size, whereas system size shows little impact on the probabilities to obtain other regular structures.

While autocatalysis and degradation stabilize clan structures, random ligation introduces fluctuations in the sys-

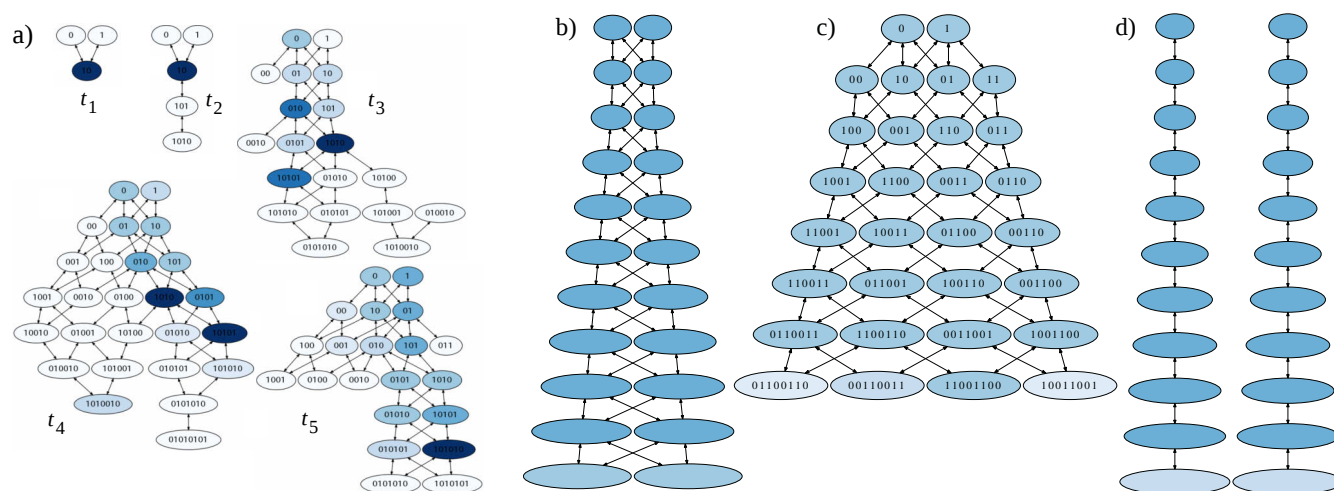


Figure 1: Representatives of the three most prominent stable cooperative structures (b)-(d) found in stochastic simulation at  $t = 100$ , together with snapshots (a) of the evolutionary formation of structure (b). More than 4000 simulations with parameters  $c_0 = 1$ ,  $c_1 = 10^{-10}$ ,  $c_2 = 10^{-7}$  and initial condition  $x_0 = x_1 = 100,000$  are performed. Graphs (b)-(d) show the average of all respective members of the three biggest clusters. Results have been aggregated using single-linkage hierarchical clustering using cosine distance. Nodes represent strings with molecular occupancy greater than or equal to 100, where darker color means larger population. Links among non-adjacent layers are omitted for clarity. 44% of the simulations resulted in a bootlace structure generated by the chiefs containing the motif 10101010 (b); another 19% of the simulations produced a pinecone structure with chiefs containing the motif 00110011 (b); and another 2% of the simulations generated the two-tower structure (d) with chiefs 00000000 and 11111111. The shown structures cover more than 65% of the simulations.

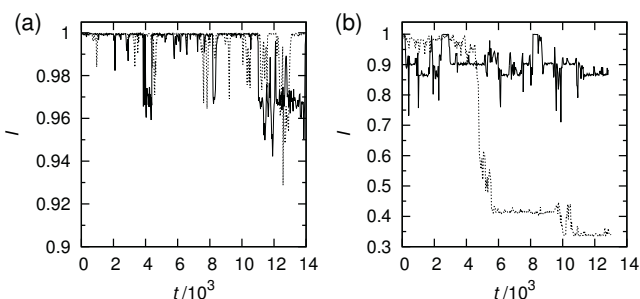


Figure 2: Fluctuations seen in the cosine similarity between an initial structure at  $t = 0$  and the one at  $t$  for (a) bootlace (solid line) and twotowers (dotted line), and (b) different pinecone structures. Note different y-axis scales.

tem. Fig. 2 shows fluctuations observed in populations drawn from the most prominent selected clan structures. While “bootlace” and “twotowers” structures (Fig 1.b and d) are relatively robust against fluctuations and reestablish the initial population after perturbations, “pinecone” structures (Fig 1.c) are more susceptible to randomness and the populations move between several metastable states which correspond to the four possible competing chiefs seen in Fig 1.c. This demonstrates how random events through selection generate punctuated equilibria in the evolutionary dynamics.

The dynamics and evolutionary potential of these systems with open boundary conditions (chemostat dynamics) remain to be studied, as well as the impact of a chemically more realistic, but also more complex, template complementarity based matching with possible sequence overhang.

## Acknowledgements

Funding for this work is provided in part by the Danish National Research Foundation and the EC sponsored projects MATCHIT and MICREAgents.

## References

- Farmer, J. D., Kauffman, S. A., and Packard, N. H. (1986). Autocatalytic replication of polymers. *Physica D*, 22:50–67.
- Gillespie, D. T. (1977). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comp. Phys.*, 22(4):403–434.
- Hordijk, W., Kauffman, S. A., and Steel, M. (2011). Required levels of catalysis for emergence of autocatalytic sets in models of chemical reaction systems. *Int J Mol Sci*, 12(5):3085–101.
- Kauffman, S. A. (1986). Autocatalytic sets of proteins. *J. Theor. Biol.*, 119:1–24.
- Rasmussen, S. (1985). PhD thesis, Technical University of Denmark, Lyngby.
- Rasmussen, S. (1989). Toward a quantitative theory of the origin of life. In Langton, C., editor, *Artificial Life*, pages 79–104. Addison-Wesley.

# On the preservation of limit cycles in Boolean networks under different updating schemes

Gonzalo A. Ruz<sup>1</sup>, Marco Montalva<sup>1</sup> and Eric Goles<sup>1</sup>

<sup>1</sup>Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Av. Diagonal Las Torres 2640, Santiago, Chile  
gonzalo.ruz@uai.cl

## Abstract

Boolean networks under different deterministic updating schemes are analyzed. It is direct to show that fixed points are invariant against changes in the updating scheme, nevertheless, it is still an open problem to fully understand what happens to the limit cycles. In this paper, a theorem is presented which gives a sufficient condition for a Boolean network not to share the same limit cycle under different updating modes. We show that the hypotheses of the theorem are sharp, in the sense that if any of these hypotheses do not hold, then shared limit cycles may appear. We find that the connectivity of the network is an important factor as well as the Boolean functions in each node, in particular the *XOR* functions.

## Introduction

Boolean networks were introduced by S. Kauffman (Kauffman, 1969) and R. Thomas (Thomas, 1973) as a mathematical model of gene regulatory networks. It has been used to model, for example, the floral morphogenesis of *Arabidopsis thaliana* (Mendoza and Alvarez-Buylla, 1998), the fission yeast cell cycle (Davidich and Bornholdt, 2008; Goles et al., 2013), and the budding yeast cell cycle (Li et al., 2004; Goles et al., 2013). Formally, let  $\mathbf{x} = \{x_1, \dots, x_n\}$  be a finite set with  $x_i \in \{0, 1\}$  for  $i = 1, \dots, n$ . Let  $N = (G, F, \pi)$  be a Boolean network, where  $G = (V, E)$  is a digraph;  $V$  being the set of  $n$  nodes and  $E$  the set of edges.  $F$  is a Boolean function,  $F : \{0, 1\}^n \rightarrow \{0, 1\}^n$  composed of  $n$  local functions  $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$ . Each local function  $f_i$  depends only on the variables belonging to the neighborhood  $V^-(i) = \{j \in V | (j, i) \in E\}$ . The indegree of vertex  $i$  is  $|V^-(i)|$ , and  $\pi$  is an arbitrary order to update the nodes  $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ . For example, the parallel or synchronous updating mode (or scheme) has  $\pi(i) = 1$  (every node is updated at the same time), whereas, for the sequential one,  $\pi$  is a permutation. A combination of the parallel and the sequential updating mode is the block-sequential where the set of nodes, for a given sequence, is partitioned into blocks. The nodes in a same block are updated in parallel, but blocks follow each other sequentially. Overall, there are an exponential number of updates. In fact,

if the network has  $n$  nodes, the number of updates is given by Demongeot et al. (2008):

$$T_n = \sum_{k=0}^{n-1} \binom{n}{k} T_k, \quad T_0 = 1.$$

Without loss of generality,  $f_i(\mathbf{x}) = f_i(x_1, \dots, x_n)$  will be used sometimes, although it should be clear to the reader that the local function really depends only on the variables in the neighborhood. Since the updating schemes are repeated periodically and the hypercube is a finite set, the dynamics of the network converges to attractors which are fixed points, i.e vectors such that  $x_i = f_i(\mathbf{x})$  for any  $i$ , or limit cycles, defined by  $x_i^{t+p} = x_i^t$  for  $i = \{1, \dots, n\}$ , where  $p > 1$  is the period. In this paper we will consider limit cycles that have *non-constant* values (a constant node does not change its value during the limit cycle) within it <sup>1</sup>.

One of the first to compare updating modes was F. Robert (Robert, 1986) for the parallel and sequential update. More recently, the robustness of such networks related to changes in the updating modes have been studied in Goles and Salinas (2008), where the authors prove that networks with monotonic loops<sup>2</sup> can not share limit cycles between the parallel and the sequential update. Furthermore, a first step to understand the different updates was done in Elena (2009); numerical experiments, under small threshold networks ( $n = 3$ ) were carried out in order to exhibit the different dynamics for every updating mode. Also, theoretical tools were developed in order to classify dynamics under different updating modes as well as to build efficient algorithms (Aracena et al., 2009; Montalva, 2011). In Goles and Noual (2012) a theoretical study of the dynamics of disjunctive networks under all updating schedules was presented. In Ruz and Goles (2013), results from reverse engineering synthesizing threshold Boolean networks with predefined limit cycles, showed that shared limit cycles, of length two, from parallel to sequential updates, were obtained for networks with indegree 3 and indegree 5.

<sup>1</sup>If there is a constant node, one may consider a new network, smaller than the original one, with non-constant nodes.

<sup>2</sup>A loop is a self connected node.

In this work, we characterize a class of Boolean network such that, two different updating modes (or a class of them) do not share non-constant limit cycles. We will see that connectivity plays an important role in this problem, in fact, if the digraph admits nodes of indegree 3, then we may find networks such that, for a specific Boolean function, the parallel and sequential updates (among others) share non-constant limit cycles (see Fig. 1). Therefore, we restrict

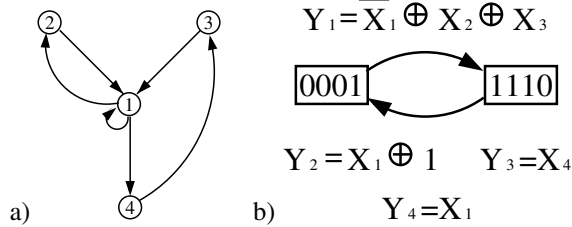


Figure 1: a) Strongly connected digraph  $G$  with  $|V^-(1)| = 3$ . b) Limit cycle shared by  $N_1 = (G, F, \pi_1)$  and  $N_2 = (G, F, \pi_2)$  where  $\pi_1 = \varphi$  is the parallel updating mode,  $\pi_2 = (2)(3)(4)(1)$  and  $F = (Y_1, \dots, Y_4)$ .

our study to digraphs of indegree at most 2. In this context, we have 16 two-input Boolean functions. It is easy to see that 14 of these functions are *canalizing* (Fogelman et al., 1982) in, at least, one input (i.e., when this input is fixed in a Boolean value  $a$ ,  $f(a, x) = f(a, \bar{x})$ ). We will prove that for the majority of the canalizing functions, one may find different updating schemes such that they share non-constant limit cycles; the exception will be the case when the function is *positive canalizing*<sup>3</sup> in its own input. In this case, the sequential and the parallel updates do not share non-constant limit cycles, which is a particular case of the result proved in Goles and Salinas (2008). Further, if the local function is *non-canalizing* (i.e., the *XOR* ( $x \oplus y = 1 \Leftrightarrow x \neq y$ ) or the *EQUIVALENCE* ( $f(x, y) = 1 \Leftrightarrow x = y$ )), we prove that, for any network of indegree at most 2 and a huge family of updating modes, the networks do not share non-constant limit cycles. In addition, for the particular case of the parallel and sequential updates, when the digraph is strongly connected we have the same result. We point out that the *XOR* and *EQUIVALENCE* functions are not monotone in any variable. It was remarked in Kauffman (1969); Walker and Gelfand (1979); Fogelman et al. (1982); Noual et al. (2012) that these kind of networks admits very long limit cycles, also, their dynamics are very sensitive to the variation of the updating mode.

<sup>3</sup>A function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is *positive canalizing* on input  $i$  if for every Boolean vector  $x$ ,  $f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \leq f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ .

## Unshared limit cycles in Boolean networks

The main result of this section is Theorem 1 that shows a sufficient condition such that a Boolean network can not share limit cycles when it is updated by different updating modes. After, we will deduce a particular case for unshared limit cycles between the parallel and any other updating mode. Next, we will show that the hypotheses of the Theorem 1 are sharp (if we change any of them, shared limit cycles may appear). Finally, we will also show that such sufficient condition is not a necessary one.

In order to prove the main result we will assume that the indegree is  $\leq 2$ , since shared cycles may appear when the indegree is  $\geq 3$ . On the other hand, we define appropriate partitions of the different updating modes.

Let  $G$  be a digraph and  $S_n$  the set of all update schedules over  $V$ . For  $i \in V$  such that  $V^-(i) = \{u, v\}$ , we define the following partition of  $S_n$ :

$$\begin{aligned} P_1(i) &= \{s \in S_n : s(u) < s(i) \leq s(v)\} \\ P_2(i) &= \{s \in S_n : s(v) < s(i) \leq s(u)\} \\ P_3(i) &= \{s \in S_n : s(i) > s(u) \wedge s(i) > s(v)\} \\ P_4(i) &= \{s \in S_n : s(i) \leq s(u) \wedge s(i) \leq s(v)\} \end{aligned}$$

and the *XOR* family  $\mathcal{F} \equiv \{x_u \oplus x_v, x_u \oplus x_v \oplus 1, x_u, x_u \oplus 1\}$ , where  $x \oplus y = 1 \Leftrightarrow x \neq y$ .

If  $i \in V$  is such that  $V^-(i) = \{u\}$ , we define the following partition of  $S_n$ :

$$\begin{aligned} P_1(i) &= P_2(i) = \emptyset \\ P_3(i) &= \{s \in S_n : s(i) > s(u)\} \\ P_4(i) &= \{s \in S_n : s(i) \leq s(u)\} \end{aligned}$$

In this case, the family is reduced to  $\mathcal{F} \equiv \{x_u, \bar{x}_u\}$ .

**Theorem 1** (Unshared limit cycles). *Let  $G$  be a digraph such that  $1 \leq |V^-(i)| \leq 2, \forall i \in V$  and consider a Boolean network  $N_1 = (G, F, \pi_1)$  that admits a limit cycle  $C$ . If every  $f_i \in \mathcal{F}$ , then, for any updating mode  $\pi_2 \neq \pi_1$  verifying that:*

$$\begin{aligned} \text{Update condition: } &\exists i \in \{1, \dots, n\}, \pi_1 \in P_4(i) \\ &\text{and } \pi_2 \in P_1(i) \cup P_2(i) \cup P_3(i), \end{aligned}$$

*the dynamics associated to the updates  $\pi_1$  and  $\pi_2$  do not share the same limit cycle  $C$ .*

*Proof.* Let  $N_1 = (G, F, \pi_1)$  be a Boolean network of  $n$  nodes such that  $1 \leq |V^-(i)| \leq 2, \forall i \in V$  and that admits a non-constant limit cycle  $C$  associated to the updating mode  $\pi_1$ . Moreover, suppose that  $f_i \in \mathcal{F}, \forall i \in \{1, \dots, n\}$ .

First, note that for this kind of functions:

$$\forall i \in \{1, \dots, n\}, \forall x \in \{0, 1\}, f_i(0, x) \neq f_i(1, x) \quad (1)$$



We denote by  $x_j^{\pi_1}(t)$  and  $x_j^{\pi_2}(t)$  the state of node  $j$  at time  $t$  for the updating modes  $\pi_1$  and  $\pi_2$  respectively.

Let  $\pi_2 \neq \pi_1$  be a given updating mode verifying (1) for some  $i \in V$  such that  $V^-(i) = \{u, v\}$  and suppose on the contrary that  $N_2 = (G, F, \pi_2)$  has the same non-constant limit cycle  $C$  of  $N_1$ , i.e:

$$\forall j \in \{1, \dots, n\}, \forall x^{\pi_2} \in \{0, 1\}^n \cap C : x_j^{\pi_2}(t) = x_j^{\pi_1}(t) \quad (2)$$

For  $i$  as above, we have that  $\pi_1 \in P_4(i)$  and the following three cases are possible:

**Case 1:**  $\pi_2 \in P_1(i)$ . Then the updates are computed by

$$x_i^{\pi_1}(t+1) = f_i(x_u^{\pi_1}(t), x_v^{\pi_1}(t)) \quad (3)$$

$$x_i^{\pi_2}(t+1) = f_i(x_u^{\pi_2}(t+1), x_v^{\pi_2}(t))$$

$$\stackrel{\text{by (2)}}{=} f_i(x_u^{\pi_1}(t+1), x_v^{\pi_1}(t)) \quad (4)$$

Because the variables in  $C$  are non-constant,  $x_u^{\pi_1}(t) \neq x_u^{\pi_1}(t+1)$ . Hence, we can assume w.l.o.g. that  $x_u^{\pi_1}(t) = 0$  and  $x_u^{\pi_1}(t+1) = 1$ . Replacing these values in (3) and (4) we have that:

$$x_i^{\pi_1}(t+1) = f_i(0, x_v^{\pi_1}(t)) \text{ and,}$$

$$x_i^{\pi_2}(t+1) = f_i(1, x_v^{\pi_1}(t))$$

Thus, due to (1) we have that  $x_i^{\pi_1}(t+1) \neq x_i^{\pi_2}(t+1)$  which contradicts (2).

**Case 2:**  $\pi_2 \in P_2(i)$ . This is similar to Case 1 but considering the calculations over  $v$ .

**Case 3:**  $\pi_2 \in P_3(i)$ . Then the updates are computed by

$$x_i^{\pi_1}(t+1) = f_i(x_u^{\pi_1}(t), x_v^{\pi_1}(t)) \text{ and,}$$

$$x_i^{\pi_2}(t+1) = f_i(x_u^{\pi_2}(t+1), x_v^{\pi_2}(t+1)) \stackrel{\text{by (2)}}{=} f_i(x_u^{\pi_1}(t+1), x_v^{\pi_1}(t+1))$$

$$= x_i^{\pi_1}(t+2) \stackrel{\text{by (2)}}{=} x_i^{\pi_2}(t+2) = \dots$$

This means that the state of vertex  $i$  must never change within  $C$ , which contradicts the assumption that each variable in the limit cycle is non-constant.

Finally, note that if  $V^-(i) = \{u\}$ , then the analysis of the case  $\pi_2 \in P_3(i)$  is similar to Case 3.  $\square$

**Corollary 1.** Let  $G$  be a strongly connected digraph such that  $|V^-(i)| \leq 2, \forall i \in V$  and consider the Boolean network  $N_1 = (G, F, \varphi)$  associated to the parallel updating mode,  $\varphi$ ,

that admits a limit cycle  $C$ . If every  $f_i \in \mathcal{F}$ , then, for any updating mode  $\pi \neq \varphi$ , the Boolean network  $N_2 = (G, F, \pi)$  does not share the same limit cycle  $C$ .

*Proof.* Let  $N_1 = (G, F, \varphi)$  be a Boolean network of  $n$  nodes that admits a non-constant limit cycle  $C$  associated to the parallel updating mode  $\varphi$ . Suppose also that  $G$  is strongly connected with  $|V^-(i)| \leq 2, \forall i \in V$ . The latter implies that:

$$1 \leq |V^-(i)| \leq 2, \forall i \in V \quad (5)$$

$$\varphi \text{ is the parallel updating mode} \Leftrightarrow \forall j \in V, \varphi \in P_4(j) \quad (6)$$

Moreover, suppose that  $f_i \in \mathcal{F}, \forall i \in \{1, \dots, n\}$ .

Let us prove the sufficient condition. Let  $\pi \neq \varphi$  be an updating mode. By (6), we have that  $\pi \in P_1(i) \cup P_2(i) \cup P_3(i)$ , for some  $i \in V$  such that  $V^-(i) = \{u, v\}$ . This, together with (5) and the assumptions made at the beginning gives us the hypotheses of Theorem 1, with  $\pi_1 = \varphi$  and  $\pi_2 = \pi$ , which guarantees that  $C$  is not a limit cycle of  $N_2 = (G, F, \pi)$ .

Finally, note that the above analysis is similar when  $V^-(i) = \{u\}$ .  $\square$

**Corollary 2.** If  $C$  is a limit cycle associated to the parallel updating mode of a Boolean network such that its digraph is a circuit, then,  $C$  is not shared by any other updating mode.

*Proof.* A circuit is a strongly connected digraph such that,  $\forall i \in \{1, \dots, n\}; |V^-(i)| = 1$  and  $f_i \in \mathcal{F} \equiv \{x_u, \bar{x}_u\}$ , where  $u \in V^-(i)$ . Thus, by Corollary 1,  $C$  is not shared by any other updating mode.  $\square$

**Remark.** In particular, we have that the parallel and the sequential updating modes do not share non-constant limit cycles.

Now, we will show the sharpness of the hypotheses of Theorem 1.

**Case 1: Not all the local functions are in  $\mathcal{F}$ .** In Fig.2a) we exhibit a strongly connected digraph  $G$  with  $|V^-(i)| \leq 2, \forall i \in V$  where the network  $N_1 = (G, F, \pi_1)$  admits a limit cycle  $C$  showed in Fig.2b) for the parallel update schedule  $\pi_1 = \varphi$  and the global function  $F = (Y_1, \dots, Y_4)$  defined in Fig.2b) as well. However, although there exists  $N_2 = (G, F, \pi_2)$  with  $\pi_2 = (2)(1, 3, 4)$  such that  $\pi_1 \in P_4(1)$  and  $\pi_2 \in P_1(1)$  as in the update condition of Theorem 1,  $C$  is shared anyway for both dynamics, because in this case, the local function  $Y_1 = X_2 \wedge X_3$  is not a XOR function. Similarly, we arrive to the same conclusion if we consider  $Y_1 = X_2 \vee X_3$  and, in general, given that every local function  $f(x, y)$  (different to a XOR function) is a combination of  $\wedge$  and  $\vee$ , we can construct other examples, as the above one, but for  $f$ .

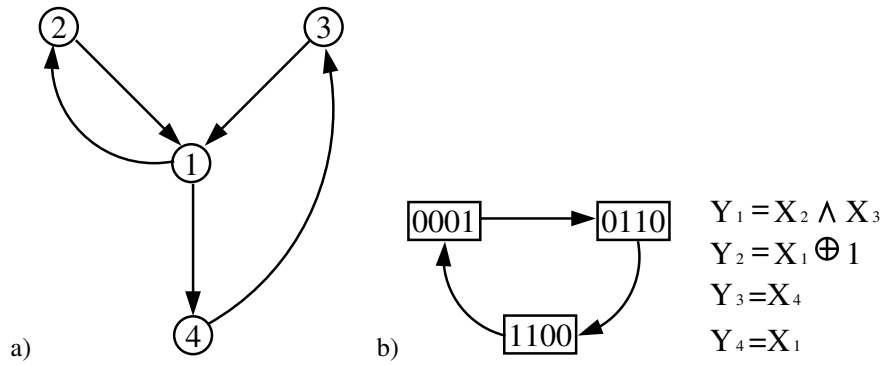


Figure 2: Example for Case 1. Note that  $Y_1 = X_2 \wedge X_3$  is the only hypothesis that does not match with Theorem 1.

**Case 2: The update condition of Theorem 1 does not hold.** First, we consider the same strongly connected digraph  $G$  of Fig.2a). The network  $N_1 = (G, F, \pi_1)$  admits a limit cycle  $C$  showed in Fig.3 for  $\pi_1 = (2)(1, 3)(4)$  and the global function  $F = (Y_1, \dots, Y_4)$  where all its components, that depend of two variables, are *XOR* functions. However,  $N_2 = (G, F, \pi_2)$  with  $\pi_2 = (3)(1, 2)(4)$  also shares  $C$ , because in this case,  $\pi_1$  and  $\pi_2$  do not verify the update condition of Theorem 1. We remark that it is also possible to show an example for each case not included in the above condition.

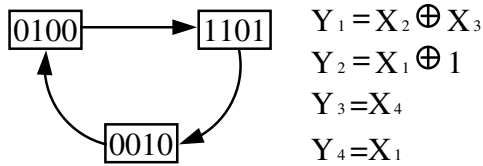


Figure 3: Example for Case 2. Note that  $\pi_1 = (2)(1, 3)(4)$  and  $\pi_2 = (3)(1, 2)(4)$  do not verify the update condition of Theorem 1.

**Case 3: Accepting constant limit cycles.** Consider the strongly connected digraph  $G$  showed in Fig.4a). Note that  $|V^-(i)| \leq 2, \forall i \in V$ . On the other hand, consider the limit cycle  $C$  showed in Fig.4b) admitted by  $N_1 = (G, F, \pi_1 = \varphi)$  where its first component is constant and equal to one. Also, on this same figure,  $F = (Y_1, Y_2, Y_3)$  is defined with all its components being *XOR* functions. So, it is easy to check that  $N_2 = (G, F, \pi_2)$  with  $\pi_2 = (2, 3)(1)$  is such that  $\varphi \in P_4(1)$  and  $\pi_2 \in P_3(1)$  as in the update condition of Theorem 1, but,  $N_2$  also shares  $C$ .

**Case 4: The sufficient condition of Theorem 1 is not a necessary condition.** The sufficient condition of Theorem 1 for unshared limit cycles (i.e., to have *XOR* functions in all the nodes with indegree equal to two) is not a necessary

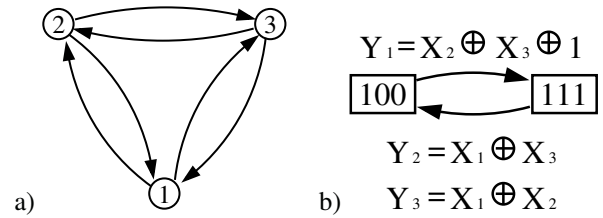


Figure 4: Example for Case 3. This constant limit cycle is the only hypothesis that does not match with Theorem 1.

condition. Consider the digraph of Fig.5a), the limit cycle  $C$  and  $F = (Y_1, Y_2, Y_3)$  of Fig.5b) verifying  $|V^-(i)| \leq 2, \forall i \in V$ . The network  $N = (G, F, \varphi)$  admits  $C$  as a limit cycle which is not shared by any other updating mode, in spite of the fact that  $Y_3 = X_1 \wedge X_2$  is not a *XOR* function.

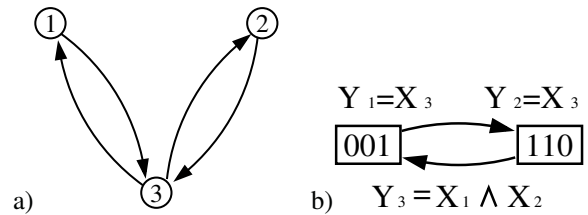


Figure 5: Example showing that the sufficient condition of Theorem 1 for unshared limit cycles is not a necessary condition.

## Simulations

In order to see how common it is to find Boolean networks (BN) that share the same limit cycles for different updating schemes (or modes), we conducted an exhaustive study for  $n = 3$  updating all the BN (graph + local functions) with

indegree at most 2, which yields 46656 BN, under all the deterministic updating schemes, i.e.,  $T_3 = 13$ . This requires order of  $10^6$  calculations approximately. Whereas to carry out the same analysis for  $n = 4$ , requires much more computational power (at least order of  $10^9$  calculations), therefore we computed only for  $n = 3$ . Also, for  $n = 3$  we are able to obtain a vast spectrum of BN necessary to study our theorem.

The 46656 BN can be divided into: 1728 *XOR* type (i.e., the functions over all its nodes are *XOR* or *XOR*  $\oplus$  1) and 44928 non *XOR* type. Also, from the 46656 BN, 33023 BN have at least one updating mode that produces at least one limit cycle (that can be either constant or non-constant). The remaining 13633 BN have only fixed points for any updating mode.

The 33023 BN can be divided into 1647 that are *XOR* type and 31376 non *XOR* type. They can also be divided in 15443 BN that have at least one updating mode that produces at least one non-constant limit cycle, the remaining 17580 BN never have non-constant limit cycles.

From the 15443 BN, 1275 are *XOR* type, and the remaining 14168 are non *XOR* type.

From the 1275 BN, 1227 share at least one non-constant limit cycle. From the 14168 BN, 10208 share at least one non-constant limit cycle.

Finally, we conclude that  $1227 + 10208 = 11435$ , which represent approximately 25% of the total, is the amount of BN that, regardless if they are *XOR* or non *XOR* type, share at least one non-constant limit cycle.

A summary of these results are shown in Fig. 6.

## Conclusion

In this paper we have characterized digraphs of low connectivity (indegree  $\leq 2$ ) and local Boolean functions (*XOR* or *EQUIVALENCE*) such that a relevant (actually, an exponential number) set of different updating modes do not share non constant limit cycles. Furthermore, the updating modes that can be compared according to the hypotheses of Theorem 1, conform a set that, in the restricted case of strong connectivity (Corollary 1), includes the particular cases of the parallel and serial updates studied in Goles and Salinas (2008) for Boolean networks with monotonic loops. In this context, taking into account our results and the ones obtained in Goles and Salinas (2008), we conclude that unshared limit cycles between the parallel and sequential updating modes occur in at least two situations: when the loops, if they exist, are positive canalizing or when all the local functions are in the *XOR* family  $\mathcal{F}$ .

Finally, the results presented in this paper using *XOR* functions, confirm their importance in the dynamical behavior of Boolean networks, a fact that was recently highlighted in Noual et al. (2012). In addition, our results use sharp hypotheses in the sense that we can exhibit counterexamples in every case where any of these hypotheses do not hold.

## Acknowledgements

The authors would like to thank Conicyt-Chile under grant Fondecyt 11110088 (G.A.R.), Fondecyt 3130466 (M.M.), Fondecyt 1100003 (E.G.), Basal (Conicyt)-CMM, and ANILLO ACT-88 for financially supporting this research.

## References

- Aracena, J., Goles, E., Moreira, A., and Salinas, L. (2009). On the robustness of update schedules in Boolean networks. *Biosystems*, 97:1–8.
- Davidich, M. I. and Bornholdt, S. (2008). Boolean network model predicts cell cycle sequence of fission yeast. *PLoS ONE*, 3(2):e1672.
- Demongeot, J., Elena, A., and Sené, S. (2008). Robustness in regulatory networks: A multi-disciplinary approach. *Acta Biotheoretica*, 56:27–49.
- Elena, A. (2009). *Robustesse des réseaux d'automates booléens à seuil aux modes d'itération. Application à la modélisation des réseaux de régulation génétique*. PhD thesis, Université Joseph Fourier, Grenoble, France.
- Fogelman, F., Goles, E., and Weisbuch, G. (1982). Specific roles of the different boolean mappings in random networks. *Bulletin of Mathematical Biology*, 44(5):715–730.
- Goles, E., Montalva, M., and Ruz, G. A. (2013). Deconstruction and dynamical robustness of regulatory networks: Application to the yeast cell cycle networks. *Bulletin of Mathematical Biology*, 75:939–966.
- Goles, E. and Noual, M. (2012). Disjunctive networks and update schedules. *Advances in Applied Mathematics*, 48:646–662.
- Goles, E. and Salinas, L. (2008). Comparison between parallel and serial dynamics of Boolean networks. *Theoretical Computer Science*, 396:247–253.
- Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467.
- Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. (2004). The yeast cell-cycle network is robustly designed. *PNAS*, 101:4781–4786.
- Mendoza, L. and Alvarez-Buylla, E. R. (1998). Dynamics of the genetic regulatory network for arabidopsis thaliana flower morphogenesis. *J Theor Biol.*, 193:307–319.
- Montalva, M. (2011). *Feedback set problems and dynamical behavior in regulatory networks*. PhD thesis, Universidad de Concepción & Université Joseph Fourier (Grenoble I), Concepción (Chile) & Grenoble (France).
- Noual, M., Regnault, D., and Sené, S. (2012). Boolean networks synchronism sensitivity and xor circulant networks convergence time. In *AUTOMATA 2012*.
- Robert, F. (1986). *Discrete Iterations*. Springer-Verlag.
- Ruz, G. A. and Goles, E. (2013). Learning gene regulatory networks using the bees algorithm. *Neural Computing and Applications*, 22:63–70.

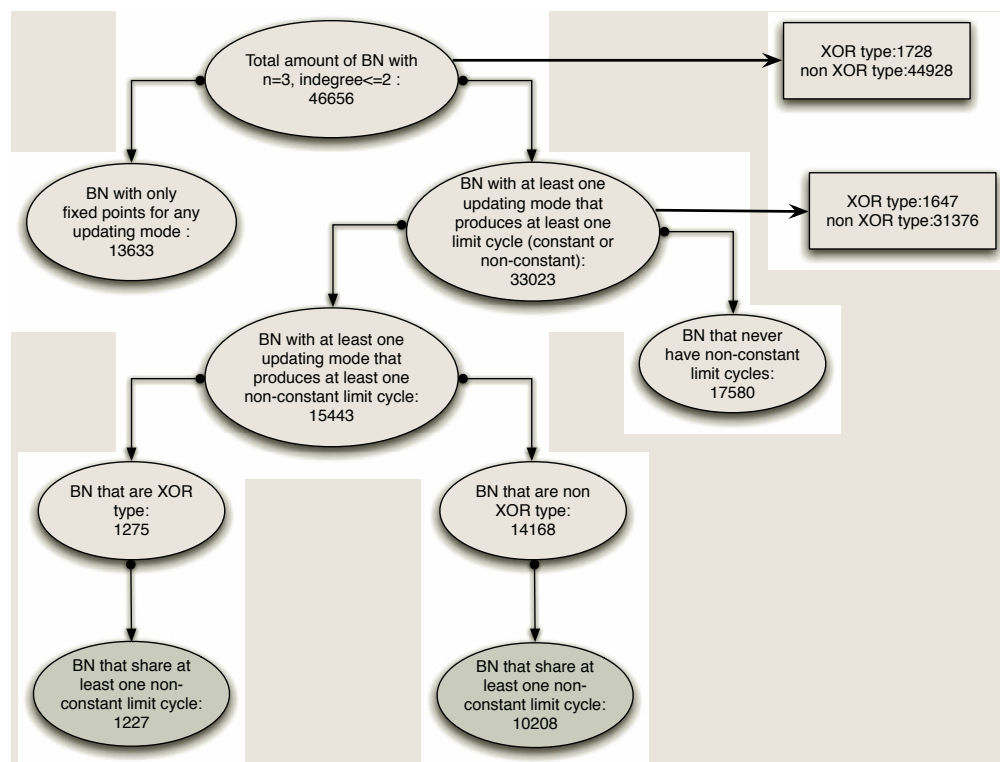


Figure 6: Summary of the results obtained by simulations that show the amount of Boolean networks (BN) that share at least one non-constant limit cycles for different updating modes.

Thomas, R. (1973). Boolean formalisation of genetic control circuits. *Journal of Theoretical Biology*, 42:563–585.

Walker, C. and Gelfand, A. (1979). A system theoretic approach to the management of complex organizations: Management by exception, priority, and input span in a class of fixed-structure models. *Behav. Sci.*, 24:112–120.



# Using Reproductive Altruism to Evolve Multicellularity in Digital Organisms

Jack Hessel and Sherri Goings

Carleton College, Northfield, Minnesota 55057  
sgoings@carleton.edu

## Abstract

The processes by which multicellular organisms first emerged from their unicellular ancestors are fundamental to the biology of complex, differentiated life forms. Previous work suggests that reproductive division of labor between specialized germ and soma cells was central to this evolution in some cases. Here, we assess the potential of the digital life platform Avida to examine the trade-off between survival and replication in multicellular organisms. Avida uses a grid of self-replicating computer programs capable of mutation and evolution to address biological questions computationally. We model our digital organisms after the Volvocales, a flagellated order of photosynthetic green algae that includes both unicellular and multicellular species. We show that, given selective pressures similar to those experienced by the Volvocales in nature, digital organisms are capable of evolving multicellularity within the Avida platform. The strategies we observed that best handled the trade-off between survival and replication involved germ cells producing sterile, somatic offspring. These strategies are similar to those observed in volvocine algae, which suggests that digital platforms, such as Avida, are appropriate to use in the study of reproductive altruism.

## Introduction

How and why multicellular organisms developed are central questions in developmental biology. In life's history, multicellularity has emerged from unicellularity on at least 25 separate occasions (Grosberg and Strathmann, 2007). The fact that this type of specialization and cooperation between cells has emerged independently and repeatedly in organisms ranging from algae to fungi suggests that this phenomenon is not a statistically unlikely event, but is the result of selective pressures experienced by various types of life. Previous theoretical and experimental work has shown multicellularity to be selectively advantageous in several circumstances (Rokas, 2008). In *Chlorella vulgaris*, for example, multicellular forms have evolved from their unicellular counterparts in the presence of a predator within 100 generations, suggesting that a multicellular existence might be advantageous to combat predation (Boraas et al., 1998). Here, we focus on the potential benefits of reproductive division of labor in multicellular forms.

Both reproduction and survival are vital for life to propagate. Differentiation between reproductive germ cells and purely functional soma cells is observed in the Volvocales, a flagellated order of photosynthetic green algae (Kirk, 2001). We chose to model our experimental parameters after the Volvocales specifically because they include multicellular organisms of varying colony size, each of which displays a different degree of complexity and specialization (Koufopanou, 1994).

The primary trade-off Volvocales address is between mobility and reproduction. An algae colony's ability to photosynthesize effectively is dependent upon its depth within the water column, and vertical traversals of entire colonies are common (Sommer and Gliwicz, 1986). A colony's capacity for mobility is primarily determined by the total functionality of its members' flagella.

In the Volvocales, however, cell division damages flagella. When a cell replicates, its flagella continue to function, but "not as strongly or as well coordinated as when [a cell is] not dividing" (Marchant, 1977). After a cell replicates several times its flagella become completely nonfunctional. The number of divisions until a cell loses all flagellar function is generally assumed to be about five (Koufopanou, 1994; Michod et al., 2006). Previous literature suggests that differentiation between germ and soma "may have evolved as a solution to this problem: by denying reproduction to some cells, a parental colony can maintain functional flagella on these cells, which will enable it to maintain its position in the water column while the rest of its cells are dividing" (Koufopanou, 1994). We refer to this constraint as the flagellation constraint.

Another consideration regards the physical volume of germ cells. Reproductive cells in differentiated volvocine colonies tend to have much greater volume than their sterile counterparts (Figure 1). This size differential results from the fact that post-embryonic cell division is not possible (Michod et al., 2006). For the purposes of designing our Volvocale-inspired digital organisms, however, establishing an association between physical volume and replication is adequate. When a colony's total volume increases, its mobil-



Figure 1: *Volvox*, a species of the Volvocales, exhibiting full germ-soma differentiation. The larger cells within the colony are germ cells, while the smaller cells are soma. Image courtesy Frank Fox of [www.mikro-foto.de](http://www.mikro-foto.de).

ity decreases because of increased mass and drag, and more total flagellation is required for colonial motion. Previous work suggests that overcoming this enlargement constraint is yet another benefit to germ-soma specialization (Michod et al., 2006).

It has been shown that reproductive altruism, cells voluntarily producing sterile, somatic offspring for the benefit of their kin, emerges in volvocine green algae. This process involves the expression of an altruistic gene within the parental cell in response to environmental cues (Nedelcu and Michod, 2006; Michod et al., 2006). We aim to implement the enlargement and flagellation constraints in a digital life platform and compare the strategies evolved by digital organisms to those evolved by the Volvocales in nature. While our model could be improved upon, our primary goal is to assess the potential of a digital life platform to address questions of reproductive altruism.

### The Avida Platform

Avida is a software platform that maintains a grid of self-replicating and mutating computer programs, and can be used to address biological questions computationally. Unlike the slow progress of organic evolution, digital evolution allows researchers to conduct experiments relatively quickly; tens of thousands of generations can be executed in a single day. Additional benefits of using a digital platform like Avida include the ability to repeat experiments exactly, specify all environmental parameters, and measure population statistics with precision. Avida is not meant to perfectly model any specific biological system, but supports the basic evolutionary processes of “replication, variation, and differential fitness.” These three conditions create an environment in which evolution will occur (Dennett, 2002). It’s

also worth noting that experimentation within Avida is “not a simulation of a particular evolutionary theory but ... an experimental study in its own right” (Ofria and Wilke, 2004).

The function of a digital organism in Avida is specified by a sequence of low-level computer instructions, in the same way that a genome composed of DNA encodes the form and behavior of a biological organism. The default Avida genomic language contains instructions that perform logical and mathematical operations, functions specific to replication such as allocating and copying memory, and flow control commands that modify the execution order of an individual’s genome. The control commands include instructions that cause small modifications to execution order such as `IF-LESS`, which may skip the immediately following instruction based on a numerical comparison between stored values, and instructions that allow for larger changes such as `MOV-HEAD`, which could cause execution to jump to an instruction anywhere in the genome.

Each Avida organism runs on its own unique set of simulated hardware, including a virtual CPU, three registers, two stacks, input/output functionality, and memory. An organism is allocated memory to hold its own genome of instructions plus extra initially “blank” memory it can use to store the genome of any child it produces. The virtual CPU executes the genome as a continuous sequence of instructions, simply starting again with the first instruction after executing the last. This hardware combined with the instructions in the default Avida language is Turing complete.

In the past, Avida has been used to examine fundamental evolutionary principles in great detail. Notable examples include an examination of the origin of complex features in biological organisms (Lenski et al., 2003) and a study of the relationship between genomic complexity and robustness (Lenski et al., 1999). Because of Avida’s demonstrated capacity to improve our understanding of evolution in general, we believe it is a worthwhile endeavor to assess its potential to address questions of reproductive altruism.

Avida in particular is well suited for our experiments because we are interested in the specific strategies digital organisms might evolve to address the flagellation and enlargement constraints. While a mathematical model or a less complex evolutionary algorithm could offer some insight and might be easier to analyze, it likely would not yield the same depth of information. Some of the more interesting strategies we observed, for instance, would not have appeared in a simpler system.

### Avidian Life-cycle

The Avida life cycle is defined as follows. First, an organism generally executes a sequence of instructions that allocates memory for its child within its own memory space, and copies each of its instructions into this newly allocated memory. Replication is asexual, and occurs when an individual executes an `H-DIVIDE` command, creating a new organism

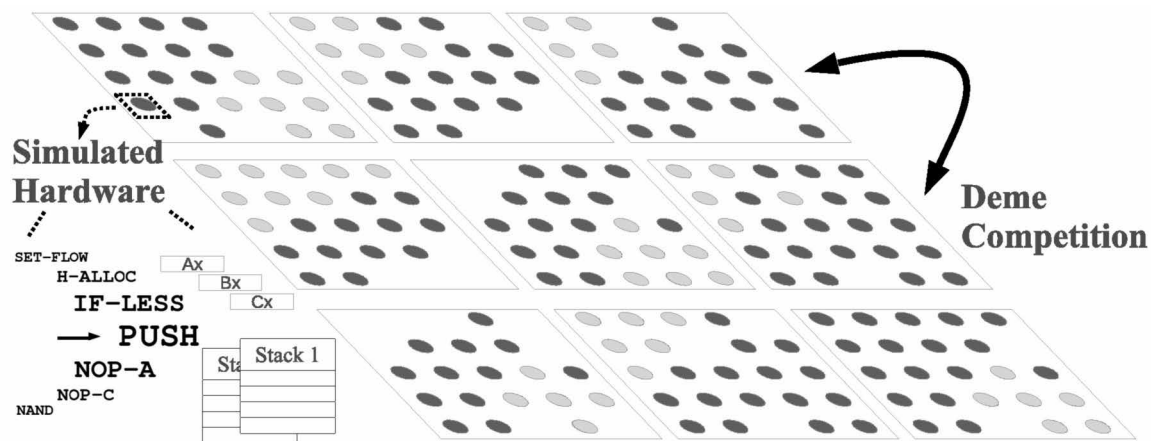


Figure 2: Avida population partitioned into demes of size 25 and an illustration of a subset of the simulated hardware

from the genome in the child memory space. The parent is reverted to its initial state with blank child memory, and begins execution again with its first instruction. In Avida several instructions in a relatively precise arrangement are required for an organism to replicate, and it is standard to seed starting populations with organisms that already have a hand-coded replication loop in their genomes. However, these instructions are subject to the same mutations and copy errors as the rest of the genome, and organisms frequently evolve modified methods of replication.

Mutation is accomplished through probabilistic instruction addition, deletion and modification within a child's genome prior to its insertion into the greater population. When an organism is copying its instructions into the memory it has allocated for its child, for instance, there is a user-defined probability that the instruction it is attempting to copy will be replaced with a random instruction from the valid set. Avida mutation rates are generally much higher than those found in the natural world; around one mutation on average per child genome is standard.

Organism death occurs either through old-age, or when an organism is overwritten by another's child. All organisms are allowed to execute the same number of instructions on average per update, though the allocation of cpu cycles is stochastic. Therefore organisms that successfully copy themselves using fewer total instructions than average generally have a higher relative fitness. However just as in the nature, many other factors affect a genome's long-term success, such as its robustness to the relatively high mutation rate.

### Multicellularity and Digital Platforms

The origins of multicellularity have previously been investigated in several contexts using artificial life models. Furu-sawa and Kaneko (1998) used an artificial chemistry model to examine multicellular emergence on a simulated two-dimensional grid. Their focus was largely on exploring the

mechanisms that cause cell differentiation during development. Other researchers have studied task-related division of labor, wherein individuals cells cooperate to perform specialized tasks efficiently, as a mechanism by which multicellularity can arise (Michod, 2007). Goldsby et al. (2010) investigated task-related division of labor within the Avida platform and found that digital organisms are capable of selecting specialized roles in groups using both spatial information and inter-organism communication.

In this paper we focus solely on the potential benefits of *reproductive*, as oppose to task-related, division of labor.

Finally, Schlessinger et al. (2006) provided an excellent investigation into the emergence of multicellularity in which they extended the Mosaic World software system to allow for optional organism aggregation into multicellular units. While the authors do include the ability for organisms to forfeit their reproductive capacity, individuals that join a multicellular unit lose their autonomy entirely, as tasks carried out by an aggregation are decided "democratically" through a poll of all constituents. Our investigation, in this context, is orthogonal, utilizing forced aggregation and optional autonomy.

### Extensions to the Avida Platform

We extend the Avida platform to incorporate flagellation and enlargement constraints similar to those exhibited by the *Volvocales* by adding two variables to each digital organism, the *flagella* value and the *physicalSize* value.

*flagella* is a numerical variable that takes on values between  $[0, 1]$ . This variable represents the functionality of a cell's flagella. When a new cell is created, we assume that its flagella are fully functional, and assign it a *flagella* of 1. If and when a cell executes a divide command, we model the flagellation constraint by decreasing this value. We decrease a cell's *flagella* value either linearly, by subtracting .25 upon replication, or exponentially, by dividing by 2 upon replication, depending on the specific experiment.

To model the enlargement constraint, we use the *physicalSize* variable, which represents the physical volume of a cell. While many species of volvocine algae replicate through a more complex process known as palintomy (Michod et al., 2006), we found a simpler binary fission model to be acceptable for our organisms.

In accordance with our binary fission model, we increment a cell's *physicalSize* value each time it executes a copy command such that *physicalSize* increases linearly from 1 to 2, until the parent has copied its entire genome to the child. Upon division, this value is reset to 1.

Because our experiments dealt with cooperation, we were particularly interested in the fitness and replication of groups of cells. The *Volvocales* ultimately specialize and become multicellular forms, so it was necessary to implement multilevel selection and evaluate groups of organisms in addition to individual organisms. We utilized Avida's population partitioning functionality to group organisms into distinct and separate subpopulations, called demes. In our experiments, each deme represented a potential multicellular organism. To evaluate and replicate demes, we modified Avida's *CompeteDemes* framework to accommodate for user-defined deme-level fitness functions and *Volvocales*-inspired deme replication.

Our *CompeteDemes* implementation computes the fitness of each deme semi-periodically and executes a fitness proportional tournament selection with five demes per tournament to determine the next generation of colonies. The timing of each *CompeteDemes* execution is offset by a random value from a uniform distribution. We implemented this deviation because, in early testing, organisms evolved an understanding of perfectly periodic timing and commonly developed undesirable behaviors.

While colonial algal reproduction is considerably more complex, we base our replication implementation on the concept that, in germ-soma differentiated species, all cells in a colony are closely related, and usually derived from the same original germ cell (Michod et al., 2006; Kochert, 1975). To replicate a colony, we select the organisms with the smallest and largest *physicalSize* values, and designate them as the founders of the appropriate demes in the next colonial generation. The cell with largest *physicalSize* is likely in mid-replication during selection, while the cell with the smallest *physicalSize* is likely not replicating during selection. By selecting the physically largest and smallest organisms, we aim to allow a single germ and a single soma cell to found colonies in the next generation, if their parental colony were germ-soma differentiated. While this process differs slightly from naturally observed reproductive mechanisms, our goal was to mimic the selective pressures experienced by *Volvocine* algae, not create a perfect model of their biology.

## Experimental Design

For our experiments, we used a population of 400 demes, each of which contained a maximum of 25 organisms. Real CPU cycles were assigned to each deme according to their living population size, and then randomly to members of that deme.

The unit of time in Avida is the "update," in which an average organism executes 30 instructions. Our experiments were run for  $10^5$  "updates," and terminated after about 10 hours. The initial seed organism executed 390 instructions before reproducing, which translates to an average of 13 updates per generation. The number of instructions executed before reproduction varied considerably between populations, as well as over time and between organisms in a single population, however all values fell within the range [200, 2000].

We designed our fitness function to reward greater total deme flagellar function, and penalize greater deme physical size. With this objective in mind, we decided to limit the space of potential fitness functions to linear combinations of these two deme-level statistics. While this limitation is somewhat arbitrary, our goal was not to perfectly model *Volvocales* colonial fitness, and the following simple, linear trade-off function worked as well as, or better than, any other in evolving populations with interesting, diverse germ-soma differentiated strategies.

$$F = \max(0, \sum \text{flagella} - \frac{\sum \text{physicalSize}}{2})$$

We also implemented a floor on the minimum number of living cells a deme could contain and still be considered viable. We assign a fitness of 0 to demes with less than 15 cells.

Mutation rates were kept at the default Avida settings; for a given command being copied, there was a .75% chance of random instruction replacement. The expected number of mutations depends upon the genomic length of an individual. For a default organism with length 100, for instance, we would expect .75 swap mutations. Additionally, for a single replication event, there was 5% chance that an instruction would be added or deleted from the child's genome upon cell division.

## Results

Before discussing the specific mechanisms digital organisms developed to accomplish germ-soma differentiation, we aim to establish that producing sterile organisms is, in general, a solution to the flagellation and enlargement constraints. We compare the final dominant deme fitnesses from our 46 exponential flagella decay trials to the proportion of somatic cells in the final population for each trial. We find a positive correlation between these variables (Figure 3). This positive correlation indicates that, in general, populations that display higher degrees of germ-soma differentiation produce more fit dominant demes.



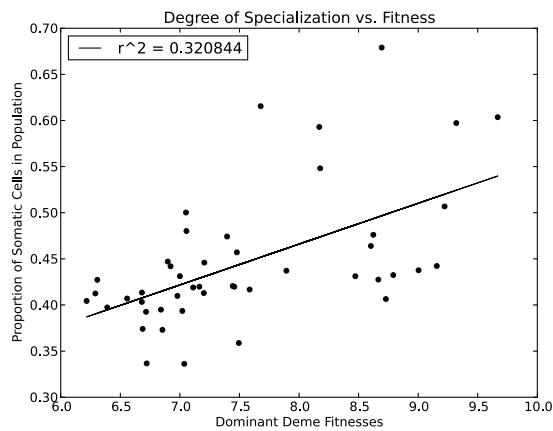


Figure 3: Least squares regression of deme fitnesses versus somatic proportion in exponential trials

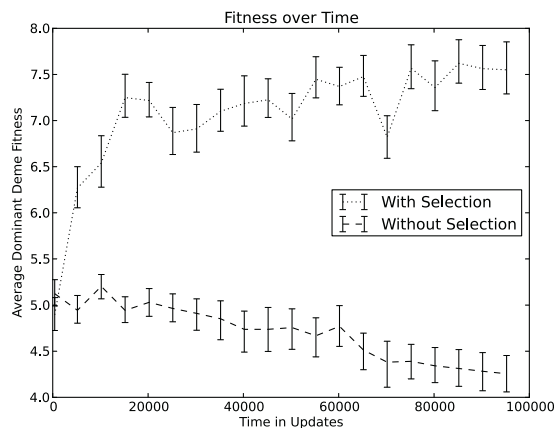


Figure 4: Average dominant deme fitness over time with bootstrapped 95% confidence intervals using  $10^5$  re-samples

Using an order statistic such as dominant deme fitness in this analysis is appropriate because our fitness function is truncated at 0; the distributions of deme fitnesses are non-normal because of this truncation. A mean, in this case, would not be an accurate reflection of overall population performance.

### Strategic Analysis

Within the exponential flagellar decay experiments, organisms evolved several different strategies, some of which resulted in germ-soma differentiation. Here, we will present these strategies in decreasing order of observed frequency. First, however, it's worth noting that within a single Avida trial, we observed some cases where multiple strategies co-existed. In these trials, it was usually the case that the deme with the highest fitness utilized one of the more complex and

interesting strategies we present here, while the most common replicator in the entire population did not.

Most commonly, no complex, germ-soma differentiated strategy emerged at all. In about 70% of our trials neither the most common nor the dominant replicator utilized a strategy other than adjusting its gestation time and genomic sensitivity to mutation, or "brittleness." The concept and prevalence of brittleness will be addressed in a later section.

The most common germ-soma differentiated strategy, which produced the dominant deme in about 13% of our trials was probabilistic replication. Organisms using this strategy produced children that were accurate copies of themselves a fixed proportion of the time. Probabilistic replicators exhibit phenotypic plasticity, which is the ability to change one's phenotype in response to environmental cues. The Avida platform inputs random numbers to cells to simulate changing environmental conditions. These inputs can be stored and operated on by organisms, if they evolve such behavior. In this case, organisms used these random numbers to determine whether or not to replicate. Those that did not replicate were basically somatic cells in the colony, with the maximum flagella value and minimum physical size. Thus the colony as a whole gained the benefits of germ-soma cell specialization though all constituent cells had identical genomes.

We also observed more complex manifestations of probabilistic replication. In one trial, for instance, a cell replicated with a probability of about 64% but did not always attempt to make an exact copy. Very rarely, this organism produced a purely somatic cell with a separate genotype. The probability of this occurrence was too low to estimate accurately, as a population of several thousand parental cells was required to observe tens of these somatic cells.

The other most common germ-soma differentiated strategy we observed involved parental organisms deterministically producing a tiered set of offspring. For example, in one of our trials, a replicative organism deterministically produced a second, different organism, and this new organism produced sterile offspring. The observed frequency of this behavior in our sample of trials was exactly the same as the probabilistic replicator.

Tiered sets of size three and four were observed, each of which contained exactly one somatic genotype and two or three germatic genotypes, all with similar gestation times. Inherently, these tiered replicative structures encode an expected proportion of sterile cells at the time of deme replication. While the best deme produced by either of these tiered strategies was produced by a three-tiered cell, the proportion of nonviable demes within the three-tiered trial was greater than the proportion of nonviable demes within the four-tiered trial ( $p < .05$ , Student's t-test) indicating that the four-tiered strategy might be more stable.

Of note was one of our trials that could be considered both a tiered and a probabilistic replicator. This particular indi-

vidual produced a tiered set of offspring, where each tier had a statistically unique, associated replicative probability that ranged from about 15% to about 25%.

The least common multicellular strategy that emerged involved cells producing both exact, germatic copies and sterile, somatic cells in deterministic sequences. This behavior was only observed in about 4% of trials. An example of an observed sequence-replicator was a cell that always produced exactly one soma before attempting to replicate itself indefinitely. We also observed genomes that encoded more complex sequences; another cell first produced a single soma, followed by a repeating sequence of one copy of itself followed by two somatic cells.

Because each of these strategies could manifest in an infinite number of ways, however, constructing a definitive ranking of strategic superiority from our data is impossible. We argue that it is only appropriate to compare the ultimate effectiveness of specific strategic manifestations, rather than the strategies themselves.

It is clear, however, that strategies which emerged in trials where selection was applied perform better than no strategy at all. When selection is not applied, mean dominant deme fitnesses are consistently lower, according to data collected from 46 trials with selection and 50 trials without selection (Figure 4).

Trials with “no selection” were still subject to some experimental constraints which selected for certain behaviors. In no selection trials, all demes were assigned identical fitnesses, making deme-level tournament selection random. From demes that were randomly selected for replication, two random organisms were selected as founders. The most common behavior that emerged within these trials was decreased gestation time. Organisms replicating increasingly quickly with no respect to the flagellation and enlargement constraints decreased average deme fitness over time.

### Brittleness

Replicative cells within trials where selection was applied also tended to evolve heightened sensitivity to lethal mutations, increasing their genotypic brittleness; germ cells that developed this behavior ultimately increased the proportion of their offspring that were sterile. This behavior was common in successful replicative organisms that did not exhibit any of the germ-soma differentiated strategies.

The Avida platform supports several types of mutation, and it's likely that cells evolved increased brittleness with respect to each. However, measurement of total brittleness is a computationally expensive process; accounting for every possible combination of legal mutations is impossible. Here, for the sake of computational complexity, we restrict our consideration to the most common mutation type: single-swap.

For each locus in a given organism's genome, each instruction within the legal set was swapped in, holding the

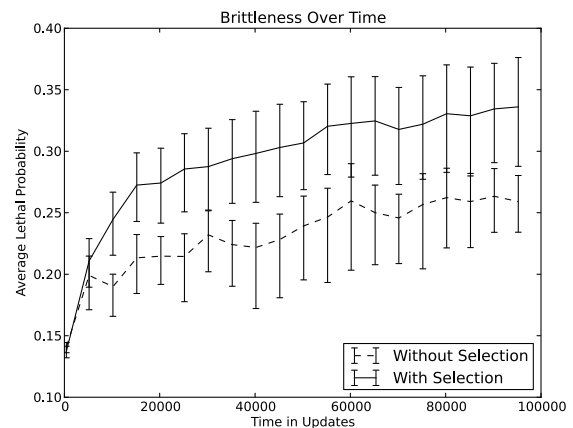


Figure 5: Average single-swap mutation brittleness over time in selection and no selection trials with bootstrapped 95% confidence intervals using  $10^5$  re-samples

rest of the genome constant. The resulting genotype was then analyzed for viability, and the fraction of single-swap mutations that rendered an organism sterile was recorded.

Here, we examine the most common replicative genotype in each of our trials and compare the effect of applying Volvocale-inspired selection on average brittleness. Even in trials where multicellular strategies emerged, however, it was usually the case that the most common replicator did not exhibit a germ-soma differentiated strategy. By examining these typically undifferentiated cells, it was our goal to establish that increased brittleness alone commonly emerges as strategy for addressing the flagellation and physical size constraints.

We find that the average brittleness of the most common replicator in trials where we applied selection was always greater than in trials where we did not (Figure 5). The difference is more consistent early in trials, indicating that increased brittleness likely emerges quickly as an “easy” solution to the flagellation and enlargement constraints prior to the emergence of more complex, differentiated strategies.

In general, the brittleness of organisms in Avida tends to increase naturally over time as genome length and organism complexity increase, which accounts for the increased brittleness we observe in the trials with no applied selection.

### Linear versus Exponential Decay

Up to this point in our analysis, we have not addressed our linear decay trials. We observe that implementing linear decay results in less germ-soma differentiation than exponential decay ( $p < .05$ , Student's t-test). Michod et al. (2006) note that a higher initial cost of reproduction yields a larger benefit from soma specialization to population viability. The Volvocales fitness can be represented in general as a multi-objective problem with an indirect relationship between via-

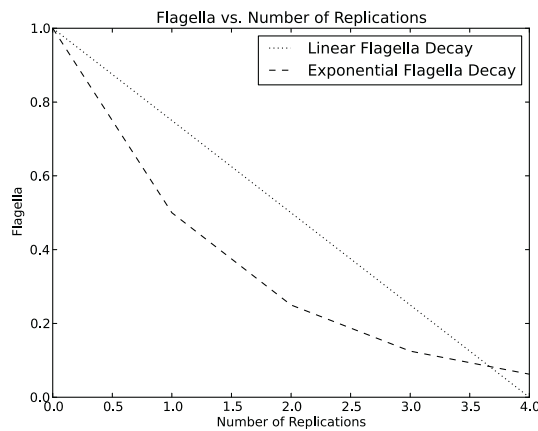


Figure 6: Flagellar function versus number of replications for exponential and linear flagella decay trials.

bility and fecundity, as a cell has limited resources to invest in each. Multi-objective problem theory predicts that if the trade-off between viability and fecundity forms a concave curve, generalists will evolve, while a convex curve will lead to specialists (Deb, 2005). Increasing the initial cost of reproduction pushes that curve to be more convex. In our experiments, an exponential decay of flagella yielded a much higher initial cost of reproduction than a linear decay, and thus more often led to the evolution of specialists (Figure 6).

## Conclusions and Future Work

We have shown that digital organisms are capable of evolving multicellularity as a solution to the flagellation and enlargement constraints within the Avida platform. The wide range of effective strategies involving germ-soma specialization we observed indicates that digital platforms can be appropriate for studying reproductive altruism. Avida, specifically, was well suited for our experiments because it offered detailed insight into novel strategies digital organisms might use to accomplish germ-soma differentiation.

In the future, we aim to improve upon our model in several ways. For example, in the Volvocales, inter-deme competition is negligible because all organisms result from the same germ cell. In our experiments, however, this type of competition exists, particularly prior to differentiation. We believe that inter-deme competition in Avida could be eliminated by implementing a more dynamic, less periodic deme replication. For instance, a deme could replicate when it contains a given number of cells, rather than at a specific time.

Another potential experimental extension would be to incorporate more physically-inspired parameters into our models. Our deme-level fitness function, for instance, might be better informed by a more careful consideration of the interplay between cell radius, volume and drag. Such consid-

erations would likely not alter our core findings, but might produce new types of organism behavior.

Finally, previous work suggests that, as colonial size increases, multicellularity becomes more advantageous (Michod et al., 2006). This theoretical result is supported by biological observation. In *Volvox*, a multicellular species of the Volvocales, colonies can contain thousands of cells (Kirk, 1998). Allowing for greater colonial size would increase the computation time required for our experiments, but would likely result in a higher proportion of trials evolving multicellularity.

## References

- Boraas, M. E., Seale, D. B., and Boxhorn, J. E. (1998). Phagotrophy by a flagellate selects for colonial prey: A possible origin of multicellularity. *Evol. Ecol.*, 12:153–164.
- Deb, K. (2005). Multi-objective optimization. In Burke, E. K. and Kendall, G., editors, *Search Methodologies*, pages 273–316. Springer US.
- Dennett, D. (2002). Encyclopedia of evolution. In Pagel, M., editor, *The New Replicators*, pages E83–E92. Oxford University Press.
- Furusawa, C. and Kaneko, K. (1998). Emergence of multicellular organisms with dynamic differentiation and spatial pattern. *Artificial Life*, 4(1):79–93.
- Goldsby, H. J., Knoester, D. B., and Ofria, C. (2010). Evolution of division of labor in genetically homogenous groups. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*.
- Grosberg, R. K. and Strathmann, R. R. (2007). The evolution of multicellularity: A minor major transition? *Annu. Rev. Ecol. Evol. Syst.*, 38:621–654.
- Kirk, D. L. (1998). *Volvox: Molecular-Genetic Origins of Multicellularity and Cellular Differentiation*. Cambridge University Press.
- Kirk, D. L. (2001). Germ-soma differentiation in volvox. *Dev. Biol.*, 238(2):213–223.
- Kochert, G. (1975). Developmental mechanisms in volvox reproduction. *Symp. Soc. Dev. Biol.*, 33:55–90.
- Koufopanou, V. (1994). The evolution of soma in the volvocales. *Am. Nat.*, 143(5):907–931.
- Lenski, R. E., Ofria, C., Collier, T. C., and Adami, C. (1999). Genome complexity, robustness and genetic interactions in digital organisms. *Nature*, 400:661–664.
- Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423:139–144.
- Marchant, H. J. (1977). Colony formation and inversion in the green alga *Eudorina elegans*. *Protoplasma*, 93(2-3):325–339.
- Michod, R. E. (2007). Evolution of individuality during the transition from unicellular to multicellular life. *Proceedings of the National Academy of Sciences*, 104(Suppl 1):8613–8618.

- Michod, R. E., Viossat, Y., Solari, C. A., Hurand, M., and Nedelcu, A. M. (2006). Life-history evolution and the origin of multicellularity. *Theor. Biol.*, 239:257–272.
- Nedelcu, A. M. and Michod, R. E. (2006). The evolutionary origin of an altruistic gene. *Mol. Biol. Evol.*, 23(8):1460–1464.
- Ofria, C. and Wilke, C. O. (2004). Avida: A software platform for research in computational evolutionary biology. *Artif. Life*, 10(2):191–229.
- Rokas, A. (2008). The origins of multicellularity and the early history of the genetic toolkit for animal development. *Annu. Rev. Genet.*, 42:235–351.
- Schlessinger, E., Bentley, P. J., and Lotto, R. B. (2006). Investigating the emergence of multicellularity using a population of neural network agents. In *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature*, pages 711–720.
- Sommer, U. and Gliwicz, Z. M. (1986). Long range vertical migration of volvox in tropical lake cahora bassa (mozambique). *Limnol. Oceanogr.*, 31:650–653.



## Some ways to see two in one

Martin Biehl<sup>1</sup> and Daniel Polani<sup>1</sup>

<sup>1</sup>University of Hertfordshire, Hertfordshire, UK  
m.biehl@herts.ac.uk

### Abstract

We present work towards clarifying whether and how the idea of agents as “subsystems” of an underlying (artificial) universe can be captured formally. For this we propose formal notions of a universe, a decomposition into subsystems and a criterion to prefer some choices of such decompositions over others. Universes are modelled by finite Markov chains, a decomposition is an information conserving set of subprocesses induced by partitions of the state space and our criterion prefers decompositions that improve predictability by minimizing stochastic interaction. Using very simple examples we find three different classes of Markov chains, with respect to their “decomposability”. Our approach also highlights the fact that the stochastic interaction of multivariate finite Markov chains crucially depends on the chosen multivariate structure of the state space.

### Introduction

#### Motivation

In this publication we address two problems, how to describe a given system as a composition of subsystems (in this paper, only two), and how to pick suitable descriptions among the possible ones. Note, when we say subsystems, we use this term at first in an intuitive way, without having a rigorous definition. Furthermore, we will use “decomposition” and “description” interchangeably as a shorthand when referring to a description as a composition of two subsystems.

The individual subsystems induced by a description should be suitable to represent both agents and environments. Here the agent and its environment could be from biological systems, robots or virtual creatures. This means the class of original systems and the subsystems themselves must be general enough to at least model these three scenarios. Additionally each description should conserve all information about the original system.

Concerning the criterion used to pick descriptions we are eventually seeking one that checks whether the two subsystems can be seen as an agent-environment pair (in other words, form a perception-action loop) and not just a pair of arbitrary subsystems. Note that such a criterion would have to be able to detect agency in the subsystems. We leave

this problem to future work. Here we are satisfied with a weaker criterion, one that can be interpreted to measure the “twoness” of the decomposition. Several concepts can be interpreted in this way: a first candidate would be the independence of the two subsystems from each other; a second related but not necessarily identical concept would be the self-determinedness or autonomy of the individual subsystems; a third would be an increase in simplicity of some kind achievable by a decomposition.

There are two scenarios where describing a system as a composition of subsystems is of interest and our approach might be applied in the future. In the first, take the perspective of an external observer of an artificial universe (e.g. a cellular automaton like the game of life). There is to our knowledge no established way to define subsystems which represent interesting entities/objects (say gliders). In artificial life the interesting entities are those that are the most life-like. The subsystems we seek should be able to represent those and the criterion we seek should pick them out. In the second scenario, take the perspective of an agent subject to a sensor input stream. If indeed decomposing the input stream makes it simpler to process, the agent could benefit from this. Additionally, it has been argued (Salge and Polani, 2011) that detecting other agents has merits in its own right. Roughly speaking, other agents produce more information relevant to an agent than is available in other parts of the environment. This suggests, that in a very general way it is advantageous for an agent to view its sensory input stream as being composed of subsystems, some agents and one environment.

#### Related work

Our work is strongly influenced by Kolchinsky and Rocha (2011). They investigate how learning a model of some given system (a multivariate Markov chain) can be improved by modeling it as a composition of independent subsystems (also Markov chains). In case of a low number of samples to learn from, they find that indeed the predictive error of the model is lowest for a composite model. Their approach is especially convincing in the scenario of an agent subject to an

input stream; the agent modeling the stream as a composite system gaining predictive power in comparison to modeling it as a non-composite one. So in this sense decomposition simplifies prediction for the agent. Note that they vary the compositions of models that are being learned not the descriptions of the original system itself. As we will see the same predictive error can also serve as a useful criterion in the latter case.

Another approach is taken by Balduzzi (2011). He tries to detect emergent subsystems (stochastic processes in general) in a given system (also a multivariate stochastic process, the game of life is used as an example). Here, the subsystems are not chosen once and stay fixed over time, but are composed out of different “units” (groups of variables) at different timesteps and can even skip timesteps altogether. The criterion used to select good choices of subsystems is dependent on how much information (in a specific sense) the different units convey about each other. While this criterion is related to how predictive one unit is of another, this work is not easily comparable with the present one. It is related mainly through the perspective of viewing subsystems as phenomena of an underlying system. This view is also applied specifically to the glider in the games of life by Beer (2013). This is an in depth study of an interesting example of a subsystem in a larger system but up to now the method depends strongly on a preidentified structure.

The system (univariate Markov chain) and subsystems (induced subprocesses) defined by Görnerup and Jacobi (2008) are also used by us. The subsystems are only investigated individually though and compositions that conserve information about the original system are not considered. As a criterion for choosing subsystems the Markov property is used. The Markov property can be interpreted as measuring self-determinedness. We will also employ it but instead of looking directly for Markovian subprocesses we focus on properties of the composition of the subsystems.

One of the authors has previously proposed emergent descriptions (Polani, 2004). Given a stochastic process, emergent descriptions are a set of stochastic processes that contain all information about the original process, are mutually independent and apart from a single one are information conserving. The subsystems we propose here necessarily fulfill the first requirement and a dynamic version of the second one; the third one is not taken into account here.

Once a system has been decomposed into subsystems there are further measures taking into account the interaction of processes that could be used to quantify the suitability of the decompositions. An example employing a framework similar to this paper would be the autonomy measures by Bertschinger et al. (2008).

## Methods

### System

As an extremely simple model for the original system we assume we are given a stationary finite univariate Markov chain<sup>1</sup>  $\{X_t\}_{t \in I}$  (we also refer to it as the universe process) defined by the transition kernel (or Markov matrix)

$$p(x'|x) := p_{x,x'} := Pr(X_{t+1} = x_{t+1} | X_t = x_t) \quad (1)$$

as the right hand side is independent of  $t$  in the stationary case. Our assumption is that the universe process should be Markov, as there is nothing external influencing it and therefore also nothing external to store information about past states in. In the case of the agent facing an input stream this assumption is of course a crude approximation. Choosing finiteness and time discreteness is done for the great reduction of technical issues compared to more general frameworks. Importantly, stationarity may be seen as an approximation, as different choices of subsystems may be most suitable at different times. We choose a univariate process because multivariate processes pre-impose a compositional structure. Moreover, in the finite case each of the variables of the multivariate process takes values in a finite state space; such a process can always be transformed into a univariate process.

### Decomposition via coordinatizations

The decompositions we propose describe the original process via a composition of processes induced on partitions of the state space. The subsystems are then those induced processes. To ensure that the original process  $\{X_t\}_{t \in I}$  can be retrieved from the subprocesses  $\{A_t\}_{t \in I}, \{B_t\}_{t \in I}$  we choose the partitions (we abuse notation and also call them  $A, B$ ) such that they form Cartesian coordinates of the state space  $\mathcal{X} = A \times B$ .<sup>2</sup> For each choice of two such partitions  $A, B$  we then have a bijective map  $f_{(A,B)} : \mathcal{X} \rightarrow A \times B$ . This map is obtained as follows, denote by  $f_A(x) : \mathcal{X} \rightarrow A$  the function mapping  $x$  to the element  $a$  (also called a *block*) of the partition  $A$  that it belongs to, and analogously for  $B$ . Then define

$$f_{(A,B)}(x) := (f_A(x), f_B(x)) = (a, b). \quad (2)$$

It is easy to see that the inverse is equal to the intersection of the blocks  $a$  and  $b$ , which is a unique state  $x$  in this case:

$$f_{(A,B)}^{-1}(a, b) := \{x : x \in a \cap b\}. \quad (3)$$

<sup>1</sup>Either we choose the index set  $I$  as the integers or we initialize the process in its stationary distribution at some time  $t = t_0$  and let it run indefinitely.

<sup>2</sup>This condition ensures for the purpose of this paper the more general requirement, that the subprocesses lose no information about the original process, i.e.  $H(X_t | A_t, B_t) = 0$  at all times  $t$ , where  $H$  denotes the Shannon entropy.

The fact that  $\mathcal{X} = A \times B$  ensures that there is always exactly one  $x$  in this intersection. For the rest of this paper we will refer to such a pair of partitions as a *coordinatization*.

In this way we obtain an alternative description of the original process  $\{X_t\}_{t \in I}$  and can set:

$$p(a', b' | a, b) := p(f_{(A,B)}^{-1}(a', b') | f_{(A,B)}^{-1}(a, b)). \quad (4)$$

Note that in general the induced subprocesses interact with each other, so that we obtain two interacting stochastic processes. This seems suitable for representing an agent-environment pair which was one of our requirements. At least it is not uncommon to model such interaction in this way; often in the context of perception-action loops (Klyubin et al., 2004; Bertschinger et al., 2008; Ay et al., 2012).

To generate coordinatizations, we note that each map  $f_{(A,B)}$  can be visualized as a way of filling an  $|A| \times |B|$  grid with the states of  $\mathcal{X}$ . We will call this grid the *coordinatization grid*. Then rows correspond to the blocks of  $A$  and columns correspond to the blocks of  $B$ . For example, if  $f_{(A,B)}(x) = (a, b)$ , the state  $x$  will appear in the grid at the intersection of row  $a$  and column  $b$ . Each coordinatization then corresponds to coordinatization grid and vice versa.

We note though that if we rename the blocks in  $A$  ( $B$ ) this has no influence on the properties of the process induced on  $A$  ( $B$ ) or on the partitions they represent. Renaming the blocks in  $A$  ( $B$ ) corresponds to permuting rows (columns) in the coordinatization grid. We therefore can reduce the number of coordinatizations that we have to investigate by choosing only one of each set of coordinatizations that can be obtained via row or column permutations of their associated grids. This can be achieved for example by always mapping state  $x = 1$  to the top left corner; mapping states to the top row such that their values are increasing to the right; and to the leftmost column such that their values are increasing downwards. Instead of  $|\mathcal{X}|!$  ways of mapping the states to the grid (and a resulting  $|\mathcal{X}|!$  coordinatizations) there are only  $|\mathcal{X}|/|A|!|B|!$  possibilities of this kind. This is still too big a number to check for larger systems, but we can already obtain some interesting insights from analysing small ones.

### Coordinatizations and modularizations

Coordinatization of a univariate process as defined above results in a bivariate process. The method could easily be extended in order to obtain multivariate processes with 3, 4, 5... subprocesses by using 3, 4, 5, ...-dimensional coordinatization grids. In this sense any finite multivariate process is already a generalized coordinatization of some underlying univariate finite process. Given a multivariate process of  $k$  variables it is possible to combine multiple of those variables (e.g. the first few in one group the rest in another) in order to get a bivariate process i.e. a coordinatization (this is a case of what Kolchinsky and Rocha (2011) call modularization). We want to stress that by combining variables

of a multivariate process i.e. by modularization, not all possible coordinatizations can be obtained. Only if the process is viewed as univariate and the coordinatizations are constructed from there, all possible ones are obtained. In other words modularizations result in only a subset of the possible compositional structures. This can be seen when considering a bivariate process as given, combining variables further is then not possible anymore, so only one coordinatization (the given one) can be obtained by modularization. Yet if the process is defined on state space  $\mathcal{X}$  and any given structure is ignored, there are  $|\mathcal{X}|!/u!v!$  possible ways of mapping the states in  $\mathcal{X}$  into a  $u \times v$  coordinatization grid and as many coordinatizations for each pair  $u, v$  of factors of  $|\mathcal{X}|$ .

### Criterion

To choose the most suitable among the coordinatizations we calculate the *stochastic interaction* (Ay and Wennekers, 2003)  $I_{(A,B)}$  (Eq. 5 below) with respect to the coordinatization  $(A, B)$  and look for those that minimize it. This can be motivated for both scenarios, the external observer and the agent perspective.

In case of the external observer, note that stochastic interaction measures in a specific sense in how far the system dynamics is more than a composition of independent subprocesses. Conversely, a low stochastic interaction indicates the composition of independent subprocesses. For a coordinatization a low stochastic interaction can then be interpreted as a measure of “twoness”.

Next take the perspective of an agent, who assumes that its input process is a composition of independent subprocesses and models it as such. If the predictive error can be reduced by this assumption the agent would have a good reason to make it. As Kolchinsky and Rocha (2011) have argued, the predictive error of a model is the sum of two terms, the error due to the assumed composition and the error due to imperfection of the learned model. The error due to the assumed composition is quantified by the stochastic interaction and is independent of other features of the learning method or model. The second term of the predictive error (or risk) then quantifies the error due to the imperfection of the parameters of the composite model. This error decreases with the number of samples available to the learner. We do not focus on this second term here but want to find the coordinatization which minimizes the “baseline error” i.e. the stochastic interaction which a possible learned composite model can reach in the best case.

For a bivariate stochastic process like the coordinatizations the stochastic interaction is defined as<sup>3</sup>

$$I_{(A,B)}(X'|X) = \text{KL}[p(A', B'|A, B) || p(A'|A)p(B'|B)] \quad (5)$$

<sup>3</sup>For the general definition we refer reader to Kolchinsky and Rocha (2011) and Ay and Wennekers (2003).

$\text{KL}[\cdot|\cdot]$  denotes the Kullback-Leibler divergence, which is in this case defined as

$$\text{KL}[p(A', B'|A, B) || p(A'|A)p(B'|B)] := \sum_{a,b} p(a,b) \sum_{a',b'} p(a',b'|a,b) \log \frac{p(a',b'|a,b)}{p(a'|a)p(b'|b)}.$$

The marginalised transition kernels  $p(a'|a), p(b'|b)$  are calculated from the transition kernel of the original process and a given starting distribution (e.g. the stationary distribution)  $p(x)$ :

$$p(a'|a) := \frac{\sum_{x': f_A(x')=a', x: f_A(x)=a} p(x'|x)p(x)}{\sum_{x: f_A(x)=a} p(x)}, \quad (6)$$

$$p(b'|b) := \frac{\sum_{x': f_B(x')=b', x: f_B(x)=b} p(x'|x)p(x)}{\sum_{x: f_B(x)=b} p(x)}. \quad (7)$$

### Procedure

The procedure we use for the analysis of the examples below is then the following. We start with a given transition kernel  $p(x'|x)$  (or the according Markov matrix with elements  $p_{x,x'} = p(x'|x)$ ) of a finite Markov chain with state space  $\mathcal{X} = 1, \dots, |\mathcal{X}|$ . Next, we calculate the stationary distribution  $p(x)$  as the left eigenvector  $v$  to the eigenvalue  $\lambda = 1$  of  $(p_{x,x'})$ . We then generate the candidate coordinatizations  $(A, B)$  via the coordinatization grid. Finally, among the coordinatizations obtained in this way, we look (brute force) for the coordinatizations that minimize the stochastic interaction, Eq.5, which represent arguably the most “natural” decompositions of  $\{X_t\}_{t \in I}$  into two subsystems. For those coordinatizations we also check the Markov property for each of the two induced subprocesses  $\{A_t\}_{t \in I}$  and  $\{B_t\}_{t \in I}$ . This can be done by checking that for any blocks  $a, a' \in A$  the total probability  $\sum_{x' \in a'} p(x'|x)$  to transition from any  $x \in a$  to  $a'$  is independent of  $x$  (Kemeny and Snell, 1976).

### Examples

We study the ideas discussed above in a class of simple systems.

Imagine a box (agent) moving on a  $m \times n$  grid wrapped around at the edges. The agent can only move up, down, left, or right (not diagonally), and does so with probability  $1/4$  in each direction at every timestep. This system then has  $mn$  states, so the state space  $\mathcal{X} = \{1, \dots, mn\}$  and every agent position corresponds to a state of the system. To make the explanations more transparent, we fix which position of the agent corresponds to which state  $x$ . Starting with  $x = 1$  in the top left corner, enumerate the top row and continue always from left to right with all other rows (cf. e.g. Fig.1). Note that this can be chosen arbitrarily. All that matters is how these states are mapped to the grid which defines the coordinatization. The latter is an  $|A| \times |B|$  grid, and even though the products  $|A||B| = mn = |\mathcal{X}|$

are necessarily equal,  $|A|, |B|$  can be a pair of factors of  $|\mathcal{X}|$  different from  $m, n$ . For example, let  $m = 3, n = 4$  then  $\mathcal{X} = \{1, \dots, 12\}$  and the tuples  $(|A|, |B|)$  can be any of  $(2, 6), (3, 4), (4, 3), (6, 2)$  where the last two possibilities correspond to substituting  $A$  and  $B$  and will reveal no more than the first two.

We now look at this system for different  $m$  and  $n$ .

**Let  $m = 3$  and  $n = 3$ :** The world grid then looks like this:

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |

and the state space is  $\mathcal{X} = \{1, \dots, 9\}$ . Because the states of the system are just positions on the grid, as mentioned before we label the positions with the states  $x$  in the way shown in Fig. 1. Just to give an impression, the  $|\mathcal{X}| \times |\mathcal{X}| =$

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Figure 1: World grid labelled with states  $\{1, \dots, 9\}$  in the way mentioned in the text.

$9 \times 9$  Markov matrix  $(p_{x,x'})$  of this system then looks like this

$$(p_{x,x'}) = \begin{pmatrix} 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} \\ \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} \\ \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \end{pmatrix}. \quad (8)$$

Next, we choose a factor of  $|\mathcal{X}|$  as the cardinality  $|A|$  of the partition  $A$  which will be the number of states of the induced subprocess. This determines also the cardinality of  $B$  as  $|B| = |\mathcal{X}|/|A|$ . In the present case, where  $|\mathcal{X}| = 9$  we





Figure 2: This illustrates the naive coordinatization of the  $3 \times 3$  world. We show the world grid and overlay the coordinatization, the tuples in each grid position denote  $f_{(A,B)}(x) = (f_A(x), f_B(x))$  where  $x$  is the state label fixed to the grid position as described in the text. The left (right) diagram highlights the partition  $A$  ( $B$ ) with each block  $a$  ( $b$ ) given the same color.

can only choose  $|A| = |B| = 3$  as a nontrivial factorization. Thus, the world grid and the coordinatization grid look the same. Before we calculate the natural coordinatization, let us look at a particular candidate coordinatization, which for lack of a better term we call the *naive coordinatization*.

For this, map the states  $x$  into the coordinatization grid in just the same way that we mapped them into the world grid, i.e.  $f_{(A,B)}(x) = (\lfloor x/|B| \rfloor, x \bmod |B|)$ . Then the position of the box on the grid is represented by the tuple  $(a, b)$  where  $a$  denotes the row and  $b$  the column. In Fig.2 we visualize the two partitions  $A$  and  $B$  and also indicate the resulting labelling of the states (and therefore positions) of the box.

One salient feature of this coordinatization is, that the dynamics can be specified easily:

$$p(a', b' | a, b) = \begin{cases} \frac{1}{4} & a' = a \pm 1, b' = b \\ \frac{1}{4} & a' = a, b' = b \pm 1 \\ 0 & \text{else.} \end{cases} \quad (9)$$

Surprisingly, this is not the “natural” coordinatization from our perspective here, its stochastic interaction is  $I_{(A,B)}(X'|X) = 1$ . Yet, there are two coordinatizations that reduce the stochastic interaction  $I_{(A,B)}$  to zero. For both, the two partitions  $A$  and  $B$  are shown in Fig.3. Note that we can transform them into each other by “transposing” the gridlabelling which is due to the fact that we can just exchange  $A$  and  $B$ . To get an intuition for the solutions which let the stochastic interaction vanish, note that instead of rows and columns, now the blocks divide the world grid into its diagonals. We also found that the induced processes of each of those coordinatizations are Markov.

**Let  $m = 2$  and  $n = 4$ :** Choose  $|A| = 2$  so that  $|B| = 4$ . Similarly to the  $3 \times 3$  world, the naive coordinatization leads to a stochastic interaction of  $I_{(A,B)}(X'|X) = 1$  and we find 6 different coordinatizations for which it vanishes. In all of the later, partition  $A$  consists of the blocks formed by the two diagonals “winding around” the world grid (at least they can

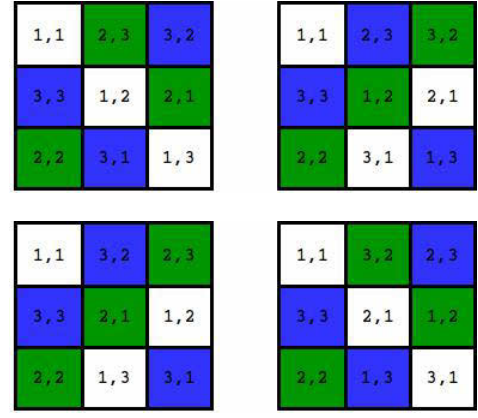


Figure 3: The two different coordinatizations of the  $3 \times 3$  world process. Each row corresponds to a different coordinatization. Again we highlight partition  $A$  ( $B$ ) in the left (right) diagram and indicate  $f_{(A,B)}(x) = (f_A(x), f_B(x))$  at each grid position. Focussing on the first coordinatization (top row), notice that when the agent is in the central position and goes up or right, the first coordinate (the block of partition  $A$ ) changes from 1 to 2 in both cases, and if it goes down or left it changes from 1 to 3 in both those cases. This means that with probabilities  $p_A(2|1) = p_A(3|1) = 1/4 + 1/4 = 1/2$  the agent switches the  $A$  blocks. Similarly, for upward or leftward movement the  $B$  block changes from 1 to 3 for downward or rightward from 1 to 2, hence  $p_B(2|1) = p_B(3|1) = 1/2$ . Done similarly for all grid positions, we see that this is the same as if we had two independent agents switching between three positions at each timestep.

always be arranged like this via permutation of the columns of the coordinatization grid) see Fig. 4. The blocks of partition  $B$  take a variety of forms. Again all subprocesses are Markov.

**Let  $m = 3$  and  $n = 4$ :** First let us choose  $|A| = 3$  so that  $|B| = 4$ . Again the naive coordinatization gives  $I_{(A,B)}(X'|X) = 1$ . The minimum stochastic interaction is non-zero though, with 6 different coordinatizations reaching a value of  $I_{(A,B)}(X'|X) \approx 0.43$  see Fig. 5. Notice, that partitions which group together diagonals of the world grid are impossible for the  $3 \times 4$  grid, because, as a diagonal winds around the grid it only self intersects after traversing the whole state space, so that there is essentially only one diagonal in each direction. Note also that all partitions  $B$  that are part of the stochastic interaction minimizing coordinatizations do not induce Markov processes, while the partitions  $A$  still do.

Now let us choose  $|A| = 2$  so that  $|B| = 6$  and the world grid and the coordinatization grid are not of the same form anymore. The coordinatizations with minimum stochastic

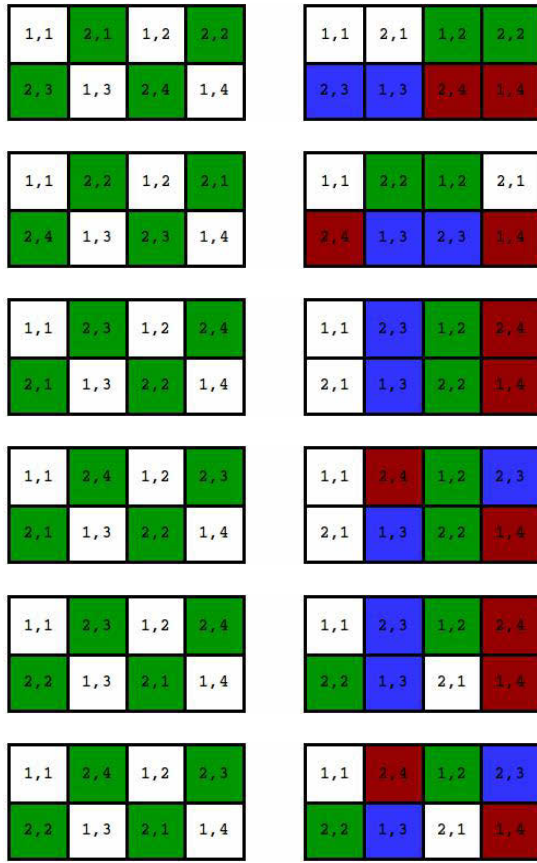


Figure 4: The coordinatizations minimizing stochastic interaction for the  $2 \times 4$  world process. The subprocesses induced by the agent dynamics on the partitions  $B$  highlighted in the right column are Markov. The same holds for the subprocess induced on the partition  $A$  shown on the left.

interaction achieve a value of  $I_{(A,B)}(X'|X) \approx 0.31$  see Fig. 6, with both  $A$  and  $B$  inducing Markov processes. It can be seen that both coordinatizations use the same patterns to cover the world grid. In fact one can obtain the other by permuting the columns of the world grid. The reason the two coordinatizations are seen as different by our algorithm is that in the coordinatization grid they cannot be transformed into one another (see Fig. 7).

**Let  $m = 2$  and  $n = 3$ :** In this case we can only choose  $|A| = 2$  and  $|B| = 3$  which leads to three different coordinatizations that achieve a minimum stochastic interaction of  $I_{(A,B)}(X'|X) \approx 0.19$ . The process induced on the partitions  $A$  is not Markov in these cases while the process induced on  $B$  is.

### Discussion

Formally, our examples have revealed three different classes of Markov chains. First, processes that allow coordinatiza-

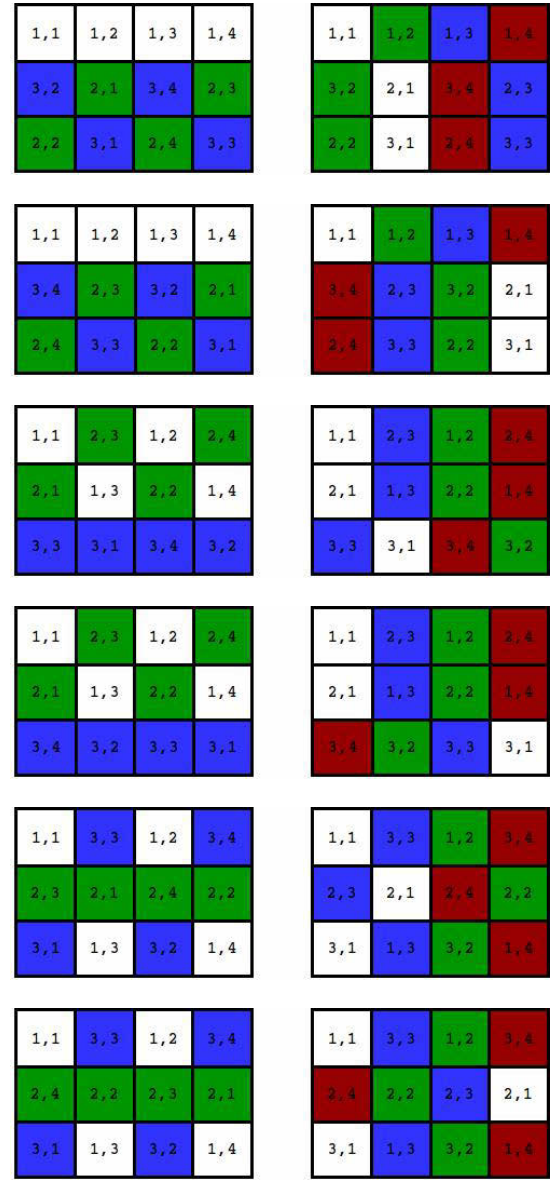


Figure 5: The coordinatizations minimizing stochastic interaction for the  $3 \times 4$  world process with  $|A| = 3$  and  $|B| = 4$ . Here the subprocesses induced by the agent dynamics on the partitions  $B$  highlighted in the right column are not Markov. Those induced on the partitions  $A$  in the left column are.

tions with vanishing stochastic interaction and Markov processes as the two subprocesses. Second, processes for which no coordinatization with vanishing stochastic interaction exists but whose coordinatizations that minimize stochastic interaction induce Markov processes (e.g. the  $3 \times 4$  world with a coordinatization grid of  $2 \times 6$ ). And third, systems where the minimum but non-zero coordinatization contains one subprocess that is not Markov ( $2 \times 3$  world). These results are summed up in the following table (“non-M” stands

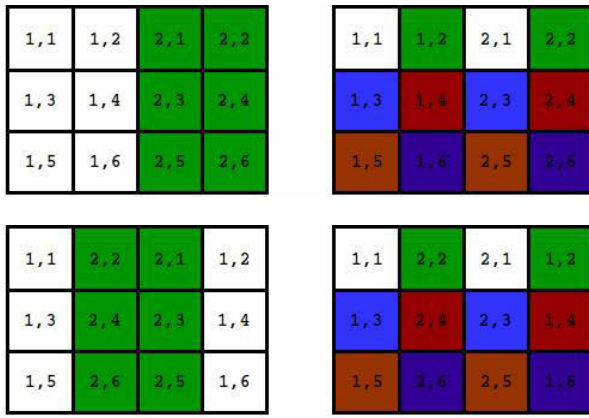


Figure 6: The coordinatizations minimizing stochastic interaction for the  $3 \times 4$  world process with  $|A| = 2$  and  $|B| = 6$  which means the coordinatization grid is different from the world grid, see Fig.7. Note that the index pairs  $f_{(A,B)}(x) = (f_A(x), f_B(x))$  reflect the  $2 \times 6$  coordinatization grid. Also note that by switching columns 2 and 4 of the grid in the first row one can obtain the grid in the second row. Also note that the induced processes of all the partitions shown here are Markov.

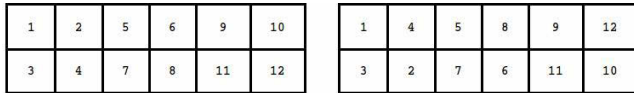


Figure 7: The coordinatization grid for the two different  $|A| = 2$  and  $|B| = 6$  coordinatizations that minimize stochastic interaction for the  $3 \times 4$  world process. Shown is the way the states  $\mathcal{X} = \{1, \dots, 12\}$  are mapped to the blocks of partitions  $A$  and  $B$ . The rows are the blocks of  $A$  and the columns are the blocks of  $B$ . For example, in the left coordinatization, state  $x = 5$  is mapped to  $(a = 1, b = 3)$ .

for not Markov and “M” for Markov):

| $m \times n$ | $ A  \times  B $ | $I_{(A,B)}^{\min}$ | $\{A_t\}_{t \in I}$ | $\{B_t\}_{t \in I}$ |
|--------------|------------------|--------------------|---------------------|---------------------|
| $2 \times 3$ | $2 \times 3$     | 0.19               | non-M               | M                   |
| $2 \times 4$ | $2 \times 4$     | 0                  | M                   | M                   |
| $3 \times 3$ | $3 \times 3$     | 0                  | M                   | M                   |
| $3 \times 4$ | $3 \times 4$     | 0.43               | M                   | non-M               |
|              | $2 \times 6$     | 0.31               | M                   | M                   |

The fact that the stochastic interaction vanishes for some coordinatizations and for some not, shows that the stochastic interaction depends crucially on the chosen coordinatization of a stationary finite Markov chain. Systems that might not seem decomposable, like the agent on the  $3 \times 3$  grid with the naive coordinatization, can in fact still allow clean decompositions like the “diagonal” coordinatization.

On the other hand the processes for which no coordinati-

zation achieves a vanishing stochastic interaction and therefore resist clean decomposition can possibly be seen as fundamentally integrated units. Here comparisons to other measures of integration e.g. Balduzzi and Tononi (2008) suggest themselves and will be pursued in future work.

More generally, our results seem to call for interaction or integration measures that are independent of the coordinatization. A dependence on the chosen cardinalities  $|A|, |B|$  (as in the  $3 \times 4$  world) may still be desirable though.

With respect to the scenario of an external observer, we could show that if an artificial universe (e.g. cellular automata) is formulated in one specific coordinatization (possibly more than two dimensional, e.g. the game of life), this coordinatization might not be the one best suited for decomposing it into subprocesses. Again recall the  $3 \times 3$  world grid; our system has actually been devised in the way of Eq. 9.

As already mentioned, the decomposition into stochastic processes seems, if only through its generality, capable of accommodating agents and environments, but our examples are inconclusive on this matter.

From the perspective of an agent subject to an input stream, it had been known that the assumption of composed processes can help reduce predictive error. Kolchinsky and Rocha (2011) have shown that given a multivariate process, assuming that groups of variables form independent processes can improve model learning. Our examples show that if the multivariate structure of the process is ignored more compositions of subprocesses can be found and might reduce the predictive error further. We can interpret this in the following ways: The multivariate structure might be imposed by the agents sensory apparatus. Then the modularizations of Kolchinsky and Rocha (2011) could be seen as the best an agent can do. From our perspective the multivariate structure is not fixed though. This corresponds to the situation where the sensory apparatus of the agent is not fixed but can still be adapted (e.g. by evolution). A third more speculative interpretation might be that the agent has to “see through” the multivariate structure and actively conceive the stochastic interaction minimizing decomposition to optimize its predictive power.

## Conclusion and outlook

Mathematically, we have proposed an approach for the decomposition of stationary finite Markov chains into pairs of subprocesses which retain all information about the original process, we refer to this as coordinatization. Minimization of stochastic interaction was used to determine “natural” coordinatizations. Three different classes of finite Markov chains showing different kinds of “decomposability” were found in the simple examples treated here.

Importantly, our approach reveals that stochastic interaction depends crucially on the chosen coordinatization. This implies that for a given multivariate Markov chain, con-

structuring coordinatizations that ignore the given multivariate structure might achieve cleaner decompositions than any grouping together of the given variables (modularization) can achieve. Such coordinatizations might in fact reduce stochastic interaction to zero. Since stochastic interaction is a lower bound for the predictive error of composite models (Kolchinsky and Rocha, 2011), our approach can in the best case be used to boost the predictive performance of such models.

In practice though the naive search method for the coordinatizations minimizing the stochastic interaction is computationally unfeasible. It remains to be seen if improved search methods can move the approach into the efficient realm.

From an artificial life perspective, we have argued for a natural choice of decompositions of a system into two subsystems for the large class of finite Markov chains. This choice can reveal ways to view and describe systems which might otherwise be overlooked. In principle the subsystems resulting from our decomposition (interacting stochastic processes) seem suitable to represent an agent and its environment or two interacting agents. Whether our criterion can be used to identify them remains an open question.

In the future we hope to extent our approach so that it can serve artificial life researchers as an analytical tool in the context of agent-environment systems or perception-action loops.

## References

- Ay, N., Bernigau, H., Der, R., and Prokopenko, M. (2012). Information-driven self-organization: the dynamical system approach to autonomous robot behavior. *Theory in Biosciences*, 131(3):161–179.
- Ay, N. and Wennekers, T. (2003). Dynamical properties of strongly interacting markov chains. *Neural Netw.*, 16(10):1483–1497.
- Balduzzi, D. (2011). Detecting emergent processes in cellular automata with excess information. *Advances in Artificial Life, ECAL*, abs/1105.0158.
- Balduzzi, D. and Tononi, G. (2008). Integrated information in discrete dynamical systems: Motivation and theoretical framework. *PLoS Comput Biol*, 4(6):e1000091.
- Beer, R. D. (2013). The cognitive domain of a glider in the game of life. *Submitted*.
- Bertschinger, N., Olbrich, E., Ay, N., and Jost, J. (2008). Autonomy: An information theoretic perspective. *Biosystems*, 91(2):331–345.
- Görnerup, O. and Jacobi, M. N. (2008). A Method for Inferring Hierarchical Dynamics in Stochastic Processes. *Advances in Complex Systems*, 11(01):1–16.
- Kemeny, J. G. and Snell, J. L. (1976). *Finite Markov Chains: With a New Appendix "Generalization of a Fundamental Matrix"*. Springer.
- Klyubin, A., Polani, D., and Nehaniv, C. (2004). Organization of the information flow in the perception-action loop of evolved agents. In *2004 NASA/DoD Conference on Evolvable Hardware, 2004. Proceedings*, pages 177–180.
- Kolchinsky, A. and Rocha, L. M. (2011). Prediction and modularity in dynamical systems. *Advances in Artificial Life, ECAL*, pages 423–430.
- Polani, D. (2004). Defining emergent descriptions by information preservation. *InterJournal Complex Systems*, (1102).
- Salge, C. and Polani, D. (2011). Digested information as an information theoretic motivation for social interaction. *Journal of Artificial Societies and Social Simulation*, 14(1):5.



# An Environmental Model of Self-Compatibility Transitions in the Solanaceae Plant Family

Paul Calcraft<sup>1</sup>, Phil Husbands<sup>1</sup>, and Andrew Philippides<sup>1</sup>

<sup>1</sup>Centre for Computational Neuroscience and Robotics, Department of Informatics, University of Sussex  
P.Calcrafft@sussex.ac.uk

## Abstract

Higher level selection processes such as species selection are not generally predicted to overpower individual selection on character traits. Goldberg et al. provide a model derived from collected life history data and argue that species selection is maintaining self-incompatibility in the Solanaceae plant family. This model applies only on the level of the species, not representing the underlying interactions between individuals and the environment. We propose a new model with environmental variation at the individual level that may explain the maintenance and frequency of loss of this character trait. We use individual based modelling techniques to explore our hypothesis, and compare it with that originally proposed. The results show alternative values required for the mutation rate to produce the species level transition frequency under the opposing models, given certain assumptions. Future work is suggested to refine the parameter relationships, test for robustness, and determine if individual models of higher complexity will exhibit similar outcomes.

## Introduction

Evolutionary questions that address multiple levels of the biological hierarchy offer a particular challenge to researchers. There is lack of consensus among biologists about the level(s) at which Darwinian natural selection should be considered to act (Okasha, 2006). This debate about the levels of selection has a complex history, marked by the group selection controversy (Wilson, 1983; Okasha, 2001), and theories of multi-level selection (Damuth and Heisler, 1988) and inclusive fitness (Hamilton, 1964; Queller, 1992).

Empirical data concerning the life history of individuals and species is in many cases insufficient to answer biological questions conclusively (Johnson and Omland, 2004; Turchin, 2013). Mathematical models have been used extensively, but they cannot capture the complexity of interactions in all cases. Individual based modelling techniques can be used to approach problems of this nature, using computer simulations of interacting systems at multiple levels. These models can expose outcomes of theoretical positions that may not be readily apparent (Di Paolo et al., 2000). Their flexibility and speed additionally allow systematic exploration of parameter spaces, testing the robustness and plau-

sibility of proposed ideas. Their potential for incorporating environmental interaction can be key in exposing the workings of natural systems (Brooks, 1991).

In this paper we model the individual interactions within the species that underlie the model of Goldberg et al. (2010). We consider individual level selection, that is, natural selection competing between individuals within a given species, to try to expose the lower level dynamics that are giving rise to the outcomes seen at the level of species. We compare two alternative formulations with different profiles of environmental interaction; the first assuming that, as proposed by Goldberg et al., species selection is acting in direct opposition to the lower level individual selection; the second introducing environmental variation. Individual level selection processes are generally considered more powerful than their higher level counterparts such as species selection (Lewontin, 1970). Species selection is therefore rarely cited as able to maintain a trait that is disadvantageous to the individual, but this has been suggested in the case of self-incompatibility in the Solanaceae (nightshade) plant family (Goldberg et al., 2010).

In the next section we discuss the evolutionary background of self-incompatibility in the context of Goldberg et al.'s work. We then introduce the theoretical concepts behind the competing evolutionary incentives. Following this, we lay out the mathematical model of fitness that will form the basis of our individual model. With the mathematical framework in place, we describe the details of the computer simulation and the relevant parameters. Finally, we go on to discuss the results.

## Self-Incompatibility and the Goldberg et al. Model

Self-incompatibility (SI) in plants is a mechanism to prevent self-fertilisation and encourage outcrossing – reproduction with those genetically dissimilar; this increases the genetic diversity of offspring (Barrett, 1988). The alternative, however – self-compatibility (SC) – can be immediately evolutionarily advantageous to individuals in SI populations, as self-fertilising (or selfing) allows plants to pass on their

genes with higher probability, and selfing plants need not rely on inbound pollen when it is scarce (Lloyd, 1992; Igic et al., 2008).

Goldberg et al. reconstructed a tree of life for the nightshade family, and found a maximum likelihood model that captures the evolution of the species in the family and their relationship with self-incompatibility and self-compatibility. The model shows that SI species have an average rate of transition to SC of 0.555 per lineage per million years, yet a proportion of SI species continues to survive over evolutionary time. This appears to be because SI species have a higher net rate of growth than SC species. This difference (a species-level advantage) is greater than the rate of transition, allowing a proportion of species to be maintained as SI ongoing, despite regular transition to SC.

In claiming that species selection is maintaining self-incompatibility, Goldberg et al. argue that individual evolutionary incentive for SC is constantly present, but the rate of arrival (and spreading) of the SC mutation in each SI population is sufficiently low to keep the transition rate averaging at this 0.555 per million years figure. This rate is low enough to allow the difference in net species growth (diversification) to be the more significant evolutionary force.

We propose a model where a background factor, a rate of occurrence of temporary environmental disruption, is the cause for a given species transition to SC. Under this model, individual selection does not constantly favour selfing, but is rather selectively neutral, or marginally favouring outcrossing. When the environmental disruption occurs, the relative fitness of selfers increases temporarily, and there is opportunity for an SC mutant to arrive and spread in the population. As long as this model can be shown to achieve the same average transition rate for reasonable sets of parameters, it may offer an alternative explanation for the maintenance of outcrossing that does not require species selection to overpower individual selection unaided.

It is important to note that while transitions from SI to SC are regular and frequent, transitions back to SI are negligibly rare (Igic et al., 2008). This may be due to the complexity of the SI system; it requires genetic coordination at many loci, so there are many points of failure (Franklin-Tong, 2008). There may also be a self-perpetuating dynamic to selfer evolution, as under certain conditions, an increase in the proportion of selfers also increases selfer fitness, making evolution back to SI once a species has been fully invaded by SC particularly improbable.

### Individual Selection Models of Selfing versus Outcrossing

A strong individual incentive for selfing is believed to be its transmission advantage, termed *automatic selection* (Fisher, 1941). Selfers have a 3:2 advantage of gene transmission, as their seeds on average contribute two gametes to the next generation to the outcrosser's one (while both averaging an

additional one through pollen) (Busch and Delph, 2012). This transmission advantage is opposed by inbreeding depression – a generalised concept for the lower average fitness of selfed progeny. Selfed progeny may have lower fitness for a number of reasons, including reduced genetic diversity, and exposition of harmful recessive alleles (Charlesworth and Charlesworth, 1987). In simple models, inbreeding depression is represented by a value  $\delta$  which is the per progeny reduced fitness for a selfed individual. If  $\delta > 0.5$ , the selfer's transmission advantage is outweighed by inbreeding depression, and outcrossing is evolutionarily preferred (Jarne and Charlesworth, 1993): 0.5 is the equilibrium level of inbreeding depression in this model of transmission advantage. This simple relation assumes that selfer pollen is just as successful as outcrosser pollen, ignoring any pollen discounting. Pollen discounting,  $0 \leq p \leq 1$ , is the reduced relative fitness of pollen spores for selfer pollen (Nagylaki, 1976).

Using Lloyd (1992)'s phenotypic model of selfing versus outcrossing, a non-zero level of pollen discounting results in a frequency dependent equilibrium value for inbreeding depression  $\delta$  (Nagylaki, 1976). That is, the maximum level of inbreeding depression required to prevent evolution to selfing varies with the proportion of selfers (explained below with eq. (3)). It can therefore provide a self-perpetuating dynamic to the evolution of selfing, as the level of  $\delta$  required to maintain outcrossing increases with the proportion of selfers, so as more selfers evolve, it becomes increasingly more difficult to maintain outcrossing. This means that if environmental circumstances are temporarily in a state that encourages evolution to SC, the proportion of outcrossers may drop below the level at which outcrossing could be maintained even once the environment returns to its previous state. This is a mechanism by which, without any assumed change in inbreeding depression, a temporary environmental disruption may cause one-way transitions to SC.

Our proposed environmental disruption is a temporary limitation of pollen dispersal in the population. This reduces inbound pollen availability to outcrossing plants by an amount  $0 \leq l \leq 1$ . This limitation also has fitness consequences for selfers, as outbound pollen from selfer plants will be less likely to reach and sire an outcrosser ovule for reproduction. The limitation still has a greater negative effect on outcrossers than selfers, as self-fertilised seeds will be unaffected by the lack of pollen dispersal, while all outcross progeny will be penalised.

### The Mathematical Model of Selfing versus Outcrossing

For general pollen limitation  $l$ , the initial fitness of outcrossers  $W_x$  and selfers  $W_s$  are:

$$W_x = 1 - l \quad (1a)$$

$$W_s = 1 - \delta \quad (1b)$$

This corresponds to inbreeding depression for selfers, and inbound pollen limitation for outcrossers. The transmission advantage also needs to be factored in, for which we adapt the model from (Lloyd, 1992). The transmission advantage is the result of an additional crossover process for outcrossers, in which their offspring have an average 50% chance of transmission of the trait carried by the inbound pollen (rather than their own) (Fisher, 1941). The outcrosser fitness is therefore scaled by  $\frac{1}{2} + \frac{1}{2}m_x$ , where  $m_x$  is the probability the mate is also an outcrosser. The complement of this amount ( $\frac{1}{2} - \frac{1}{2}m_x$ ) is added to the selfer fitness (representing those outcrosser progeny transmitting the selfer phenotype). This term, however, is scaled by the relative proportion of outcrossers to selfers and reduced by pollen limitation, as selfers only benefit as much as there are outcrosser ovules available to sire and their pollen can reach them. The comprehensive fitness equations are therefore:

$$W_x = (1 - l)\frac{1}{2}(1 + m_x) \quad (2a)$$

$$W_s = \frac{x}{1 - x}(1 - l)\frac{1}{2}(1 - m_x) + 1 - \delta \quad (2b)$$

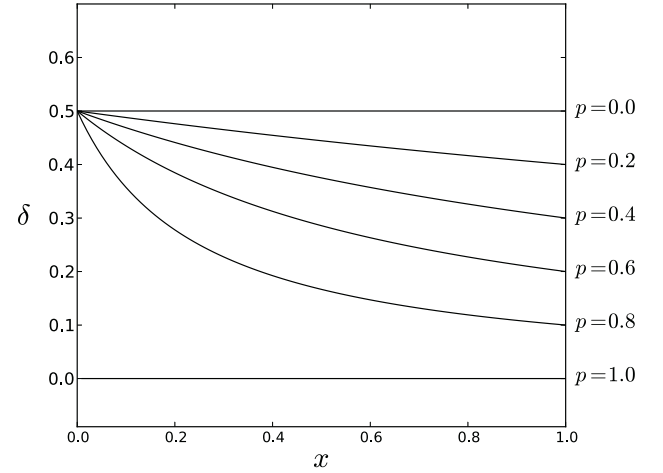
$$\text{where } m_x = \frac{x}{x + (1 - x)(1 - p)} \quad (2c)$$

Again,  $m_x$  is the probability of inbound pollen being outcrosser rather than selfer, incorporating the effect of pollen discounting  $p$ . The current proportion of outcrossers in the population is  $0 \leq x \leq 1$ .

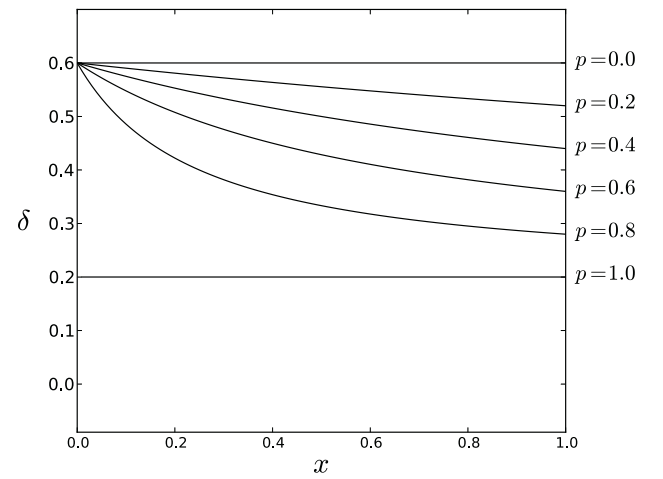
From these fitness equations we derive the level of equilibrium inbreeding depression  $\delta$ , above which outcrossing is evolutionarily preferred, and below which selfing is preferred:

$$\frac{2lp_x + (1 + l)(1 - p)}{2(p(x - 1) + 1)} \quad (3)$$

Refer to figs. 1(a) and 1(b) for an illustration of this relationship. As we can see, for  $0 < p < 1$ , the level of inbreeding depression required to maintain outcrossing increases with selfer proportion (ie. with decreasing  $x$ ); selfers have a greater advantage as the selfer proportion increases. Further, the addition of pollen limitation  $l = 0.2$  in fig. 1(b) shifts the curves upward, giving selfers a selective advantage over the  $l = 0$  condition. The curves are also contracted in the vertical ( $\delta$ ) dimension, making this difference more pronounced at higher levels of  $p$ . We use changes in the value of  $l$  to exhibit temporary environmental conditions that favour selfing.



(a) Equilibrium  $\delta$  without pollen limitation ( $l = 0$ )



(b) Equilibrium  $\delta$  under pollen limitation ( $l = 0.2$ )

Figure 1: Equilibrium inbreeding depression  $\delta$  at outcrosser proportion  $x$  for different values of pollen discounting  $p$ , with 1(b) and without 1(a) pollen limitation  $l$  (see eq. (3)).

We present two alternative models. In the first, Model A, the transition rate is caused exclusively by the arrival and fixation of the selfer mutation, under conditions that constantly favour selfing. In Model B, conditions generally favour outcrossing, but there are environmental disruptions, occurring with a certain rate  $r$ , that limit  $l$  the pollen dispersal for some duration  $d$ , during which the conditions favour selfing. This second model, the environmental model, will require a higher mutation rate than the first, as selfer morphs are only favoured by natural selection during disruptions, rather than constantly. The final point of differentiation for the two models, then, will be the mutation rate required to achieve the empirically observed transition rate, given the

background assumptions of the models.

## Methods

As stated, the target is an average transition rate of 0.555/million years. We run many repeats (500) of a single population under both conditions (original Model A and environmental disruption Model B), recording the number of years it takes to transition to selfing each time. We then take the reciprocal of the mean length of time, arriving at the average transition frequency. After fixing certain parameters of the models, we search manually for parameters that produce the target transition rate for these conditions. Using the same criterion as Goldberg et al. (2010), we conservatively classify a species as SI as long as it is not completely SC, ie. no polymorphism, approximated as less than 1% of the SI phenotype present in the population; a transition is said to occur when the outcrosser proportion goes below 0.01.

The simulation is a genetic algorithm with a single population, initially fully outcrossing ( $x = 1$ ). We use roulette selection, which is equivalent to a diffusion approximation of selection and drift (Cherry and Wakeley, 2003). The fitness values for outcrossers and selfers are as per eqs. (1a) and (1b).

Upon selection, if the phenotype is selfer, it is added to the next generation, but if outcrosser, it is combined with pollen from another plant in the population. The probability that this mate is an outcrosser, as opposed to selfer, is  $m_x$  (eq. (2c)).

The phenotype that goes into the next generation is from either the selected plant or the mate, with equal probability. This is equivalent to the average effect of crossover for outcrossing plants. The net effect of this selection and probabilistic recombination process is captured in fitness eqs. (2a) and (2b). The trait is also probabilistically mutated according to the (phenotypic, per gene per generation) mutation rate  $\mu$  before being added to the next generation.

After each generation, we check if the population has transitioned to SC ( $x < 0.01$ ) and break out of the current run if this is the case, recording the length of time that has passed. One generation is equal to one year, a working value used by other models of plants in the Solanaceae family (Vekemans and Slatkin, 1994). For a high level overview of the computer simulation's operation, refer to algorithm 1.

## Parameters

Table 1 shows the initial set of parameters for the models. The effective population size  $N_e$  for Solanaceae does vary, but 6000 is within the expected range (Richman et al., 1996). A conservative level of pollen discounting, 0.2, has been chosen initially. As explained, Model A requires that the conditions favour selfing constantly, so a value of  $\delta = 0.3$  has been chosen to fulfil this requirement (see fig. 1(a), 0.3 is below the  $\delta$  equilibrium for  $p = 0.2$ ,  $l = 0$ ). For Model B, the environmental disruption model, we need selective

```

1 for each repeat do
2   generations_until_transition = 0;
3   while outcrosser_proportion > 0.01 do
4     if disruption_generations_remaining = 0 then
5       pollen_limitation = 0;
6     else
7       disruption_generations_remaining -= 1;
8     end
9     if random() < disruption_rate {r} then
10      pollen_limitation =
11      disrupted_pollen_limitation {l_d};
12      disruption_generations_remaining =
13      disruption_length {d};
14    end
15    for population_size {N_e} do
16      roulette select an individual;
17      if individual is outcrosser then
18        pick mate according to pollen
19        frequencies {m_x};
20        crossover with mate;
21      end
22      probabilistically mutate {μ};
23      add to new generation;
24    end
25    generations_until_transition++;
26  end
27  record generations_until_transition;
28 end
29 print 1/(average(generations_until_transition));

```

**Algorithm 1:** Model algorithm

neutrality or favoured outcrossing under background pollen limitation  $l = 0$  (fig. 1(a), 0.5 is above the  $\delta$  equilibrium for  $p = 0.2$ ,  $l = 0$ ), and favoured selfing under the disruption condition (fig. 1(b), 0.5 is below the  $\delta$  equilibrium for  $p = 0.2$ ,  $l = l_d = 0.2$ ).

|          | Parameter description                | Model A | Model B |
|----------|--------------------------------------|---------|---------|
| $p$      | Pollen discounting rate              | 0.2     |         |
| $l$      | Background pollen limitation         | 0       |         |
| $l_d$    | Pollen limitation (disrupted)        | N/A     | 0.2     |
| $N_e$    | Effective population size            | 6000    |         |
| $\delta$ | Inbreeding depression                | 0.3     | 0.5     |
| $\mu$    | Mutation rate                        | *       | *       |
| $r$      | Disruption rate, /species/generation | 0       | *       |
| $d$      | Disruption duration, generations     | 0       | *       |

Table 1: Parameters of the model under Models A and B. Values to be found or manipulated are marked by \*.



## Results

In the first section we present the results from Model A, the model under which species selection directly opposes constant individual incentive for selfing, and Model B, where environmental disruptions bring about temporary individual incentive for selfing. We indicate parameter values under which these alternative low level models exhibit the empirically observed transition rate of 0.555 per lineage per million years ( $0.555\text{E-}6$ ) at the species level. We then go on to present some typical evolutionary trials, exposing the underlying selection mechanics of the models.

### Results from Models A and B

In each case, the resultant transition rate is the mean frequency of transition over 500 trials of the single population genetic algorithm. Parameter values were found by manual experimentation given the fixed values established by the model assumptions, detailed previously in table 1. The output parameter of interest is the phenotypic mutation rate required under each model to bring about the rate of transition observed by Goldberg et al..

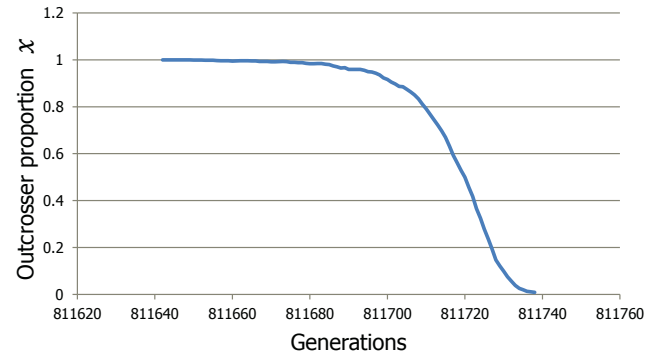
|                | $\mu$    | $r$                           | $d$                    | Transition rate |
|----------------|----------|-------------------------------|------------------------|-----------------|
| A              | 5.17E-10 | 0                             | 0                      | 0.547E-6        |
| B <sub>0</sub> | 1.33E-8  | $\langle 1\text{E-}5 \rangle$ | 5000                   | 0.563E-6        |
| B <sub>1</sub> | 2.17E-8  | $\langle 1\text{E-}5 \rangle$ | 3000                   | 0.537E-6        |
| B <sub>2</sub> | 2.17E-7  | $\langle 1\text{E-}5 \rangle$ | 500                    | 0.552E-6        |
| B <sub>1</sub> | 2.17E-8  | 1E-5                          | $\langle 3000 \rangle$ | 0.537E-6        |
| B <sub>3</sub> | 2.28E-8  | 5E-6                          | $\langle 3000 \rangle$ | 0.567E-6        |
| B <sub>4</sub> | 2.33E-8  | 1E-6                          | $\langle 3000 \rangle$ | 0.572E-6        |

Table 2: Parameters and results under Model A (original, no disruption:  $r, d = 0$ ) and B (temporary environmental disruptions:  $r, d > 0$ ). Transition rate should approximate  $0.555\text{E-}6$ . The table is grouped, where values held constant are shown in angle brackets, while others were manipulated to obtain the target transition rate. Result B<sub>1</sub> is repeated in the third group for convenient comparison.

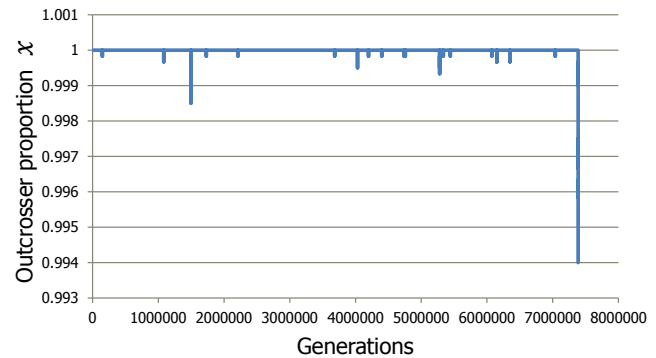
Table 2 shows that a transition rate of approximately  $0.555\text{E-}6$  can be obtained under multiple conditions; either model is able to potentially explain the empirical observations, but with a different necessary value for the mutation rate  $\mu$ . For Model A, the background assumptions are such that there is only one possible value, found to be  $5.17\text{E-}10$ . Under Model B there is more scope for interaction between the parameters during the search. Holding the disruption rate  $r$  at an average of once per 100,000 years ( $1\text{E-}5$ ), rows B<sub>0</sub>, B<sub>1</sub>, and B<sub>2</sub> show that higher mutation rates are required for shorter durations of disruption. Keeping the disruption duration  $d$  at 3000 years, we similarly see from rows B<sub>1</sub>, B<sub>3</sub>, and B<sub>4</sub> that lower values of disruption rate  $r$  require higher mutation rates, but the effect is considerably less significant.

The required mutation rate is more sensitive to the duration of the disruptions than their frequency.

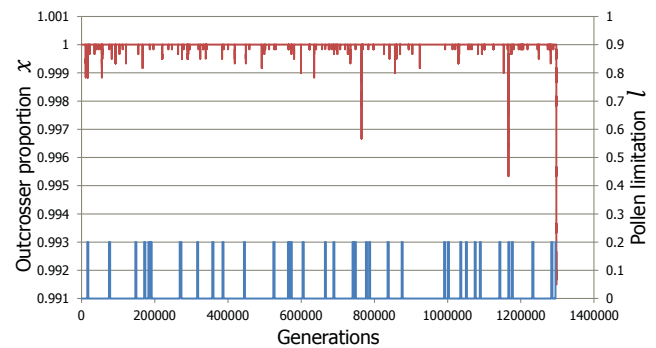
### Example Evolutionary Trials



(a) A final phase of the evolution curve once selfing manages to spread, typical under both models.



(b) An example evolution curve under Model A where SC mutations arise and are lost multiple times under drift before managing to spread and fixate. The final line down on the right continues to full selfing as in fig. 2(a). Note the scale of the y axis:  $x$  does not get below 0.9985 without SC spreading.



(c) An example evolution curve under Model B. The level of pollen limitation, alternating between 0 and 0.2 on the secondary y axis due to disruption, is also shown. Observing the scale of the primary y axis,  $x$  (the top curve) reaches below 0.9965 without SC managing to spread, lower than under Model A in fig. 2(b). The final line down on the right again continues to full selfing as in fig. 2(a).

Figure 2: Example sequences from typical evolutionary runs.

Under Model B, the temporary disruptions in the environment allow SC mutations to arise and begin to spread if they are not lost by drift, and may get further than is typical under Model A. This is illustrated in figs. 2(b) and 2(c), where the different scales of the y axis show that the outcrosser proportion can typically get slightly lower in B without a full SC invasion. This is likely due to the inconsistent selection pressure provided by Model B, as disruptions are brought in and out, shifting the balance of fitness towards and away from outcrossing over time.

Both models produce similar final phases of SC spreading to fixation, as seen in fig. 2(a), as under either model, once selfers reach a certain proportion, selection pressure becomes reinforcing and full invasion becomes highly probable.

## Discussion

Evolutionary models that consider the interaction between multiple levels of the biological hierarchy provide a complex challenge. We have taken Goldberg et al.'s species level empirical data and attempted to realise the individual level processes that give rise to the SI-to-SC species transition rate. Using a genetic algorithm and Lloyd (1992)'s phenotypic model, we discover the mutation rates required under two alternative models, given certain assumptions.

We begin to explore the conditions under which the target transition rate can be produced, and show that there seems to be scope for an environmental model to help explain the evolutionary history of SI and SC in the Solanaceae plant family. Assessing the likelihood of the presented model, or of alternative environmental variation hypotheses, will come down to the plausibility of the required mutation rates. If the mutation rate required of Model A, under the pure species selection hypothesis, is too low, this may suggest individual selection is a significant factor, mediated by environmental conditions. Our Model B presents one such possibility.

The method presented of separating out the individual selection process from the species level process may be applicable to other questions regarding multi-level selection processes. By starting with empirical evidence at the species level and reverse engineering the individual selection pressure using established models, we can explore the real world parameter ranges required to meet alternative theories. These parameters can then hopefully be subject to empirical test, to observe which model obtains.

We did not have time to perform more comprehensive parameter sweeps to provide a robustness analysis. Investigating the relationships between the sets of parameters may prove fruitful as well.

In future work, alternative theories of environmental variation should be explored. In the first instance, an alternative take on Model B would be to have pollen limitation  $l$  vary continuously in the background, rather than being manipulated by binary disruption events. It may be that gradual or

shallower yet longer dips in dispersal can produce similar rates of transition, for example. More complex models of inbreeding depression and pollen discounting should also be incorporated, as unforeseen interactions between environmental variation and fitness over time may be exhibited.

In summary, we have examined within-species dynamics, under individual selection, that can account for the species level rate of transition that has been empirically observed. Given certain conditions, we obtained the values necessary for the mutation rate to explain the data under two alternative models. Individual based modelling techniques were effectively employed, enabling the analysis of these stochastic models under environmental interaction. By attempting to establish the details of the biological interactions below the species level, we indicate parameter values that may support or reject the original species selection hypothesis.

## References

- Barrett, S. C. (1988). The evolution, maintenance, and loss of self-incompatibility systems. *Plant reproductive ecology: patterns and strategies*, pages 98–124.
- Brooks, R. A. (1991). Intelligence without representation. *Artificial intelligence*, 47(1):139–159.
- Busch, J. and Delph, L. (2012). The relative importance of reproductive assurance and automatic selection as hypotheses for the evolution of self-fertilization. *Annals of botany*, 109(3):553–562.
- Caballero, A. et al. (1994). Developments in the prediction of effective population size. *Heredity*, 73(6):657–679.
- Charlesworth, D. and Charlesworth, B. (1987). Inbreeding depression and its evolutionary consequences. *Annual review of ecology and systematics*, 18:237–268.
- Cherry, J. L. and Wakeley, J. (2003). A diffusion approximation for selection and drift in a subdivided population. *Genetics*, 163(1):421–428.
- Damuth, J. and Heisler, I. L. (1988). Alternative formulations of multilevel selection. *Biology and Philosophy*, 3(4):407–430.
- Di Paolo, E. A., Noble, J., and Bullock, S. (2000). Simulation models as opaque thought experiments.
- Fisher, R. (1941). Average excess and average effect of a gene substitution. *Annals of Human Genetics*, 11(1):53–63.
- Franklin-Tong, V. E. (2008). *Self-incompatibility in flowering plants: evolution, diversity, and mechanisms*. Springer.
- Goldberg, E. E., Kohn, J. R., Lande, R., Robertson, K. A., Smith, S. A., and Igić, B. (2010). Species selection maintains self-incompatibility. *Science*, 330(6003):493–495.
- Hamilton, W. D. (1964). The genetical evolution of social behaviour. i. *Journal of theoretical biology*, 7(1):1–16.
- Holsinger, K. (1991). Mass-action models of plant mating systems: the evolutionary stability of mixed mating systems. *American Naturalist*, pages 606–622.

- Holsinger, K. (1996). Pollination biology and the evolution of mating systems in flowering plants. *Evolutionary biology*, 29.
- Igic, B., Lande, R., and Kohn, J. R. (2008). Loss of self-incompatibility and its evolutionary consequences. *International Journal of Plant Sciences*, 169(1):93–104.
- Jarne, P. and Charlesworth, D. (1993). The evolution of the selfing rate in functionally hermaphrodite plants and animals. *Annual Review of Ecology and Systematics*, pages 441–466.
- Johnson, J. B. and Omland, K. S. (2004). Model selection in ecology and evolution. *Trends in ecology & evolution*, 19(2):101–108.
- Lande, R. and Schemske, D. W. (1985). The evolution of self-fertilization and inbreeding depression in plants. i. genetic models. *Evolution*, pages 24–40.
- Lewontin, R. C. (1970). The units of selection. *Annual Review of Ecology and Systematics*, 1:1–18.
- Lloyd, D. G. (1992). Self-and cross-fertilization in plants. ii. the selection of self-fertilization. *International Journal of Plant Sciences*, pages 370–380.
- Nagylaki, T. (1976). A model for the evolution of self-fertilization and vegetative reproduction. *Journal of Theoretical Biology*, 58(1):55–58.
- Okasha, S. (2001). Why won't the group selection controversy go away? *The British journal for the philosophy of science*, 52(1):25–50.
- Okasha, S. (2006). *Evolution and the Levels of Selection*, volume 16. Clarendon Press Oxford.
- Queller, D. C. (1992). Quantitative genetics, inclusive fitness, and group selection. *American Naturalist*, pages 540–558.
- Richman, A. D., Uyenoyama, M. K., and Kohn, J. R. (1996). Allelic diversity and gene genealogy at the self-incompatibility locus in the solanaceae. *Science*, 273(5279):1212–1216.
- Turchin, P. (2013). *Complex population dynamics: a theoretical/empirical synthesis (MPB-35)*, volume 35. Princeton University Press.
- Vekemans, X. and Slatkin, M. (1994). Gene and allelic genealogies at a gametophytic self-incompatibility locus. *Genetics*, 137(4):1157–1165.
- Wilson, D. S. (1983). The group selection controversy: history and current status. *Annual review of ecology and systematics*, 14:159–187.

# Family Bird: A Heterogeneous Simulated Flock

Jure Demšar<sup>1</sup> and Iztok Lebar Bajec<sup>1</sup>

<sup>1</sup>Faculty of Computer and Information Science, Tržaška cesta 25, 1000 Ljubljana, Slovenia

## Abstract

Since the phenomena of bird flocking is so fascinating, there is no shortage of computer models that try to visualize this mesmerizing spectacle. However, the quality of artificially modelled flocks is currently not on par with their counterparts in nature. We believe the main reason for this lies in the homogeneous structure of flocks in computer models. In this article we show how just a pinch of heterogeneity can increase the repertoire of displayed behaviours.

## Introduction

At first glance the mesmerizing phenomena of bird flocks and fish schools appear very complex, but according to existing literature the underlying principles may be quite simple. In 1980s two groups of researchers working independently showed that a flocking-like behaviour can be produced in computer simulations if artificial animals (*animats*) follow a few simple rules (Reynolds, 1987; Heppner and Grenander, 1990). Reynolds introduced three drives – *cohesion*, *separation* and *alignment*. Cohesion drives the observed animat to stay close to its neighbours; separation forces it to avoid collisions by steering away from animats that are perceived as too close; alignment imitates the desire to synchronize speed and heading with nearby animats.

The main problem of models that use these three (or similar) rules is that the appearance of the displayed behaviour is far from the mind-blowing spectacle one might admire in nature. The behaviour of computer produced flocks was very rigid and stereotyped. Many models thus introduced mechanisms that induce some randomness to the motion of the animats (Heppner and Grenander, 1990; Vicsek et al., 1995; Couzin et al., 2002; Hildenbrandt et al., 2010). The authors defend this approach by saying these mechanisms simulate wind gusts, obstacles and other random factors. We disagree with their reasoning since obstacles do not just appear at random, and wind gusts are not random per individual. Even with the addition of randomness the result was nowhere near as breathtaking as flocking in nature; the flocks in computer models seldom split into smaller flocks that rejoin after manoeuvring on their own for a while. We believe the main

deficiency of artificial flocks is their assumption that all the birds in the flocks have the same characteristics. The birds in natural flocks often differ in size, gender, age, and even species (Lebar Bajec and Heppner, 2009; Jolles et al., 2013). As shown in (Jolles et al., 2013) the flock's structure and social relationships between individuals greatly impact the behaviour of the flock. To bring computer models closer to flocks in nature, we developed a heterogeneous model that includes social relations.

## Methods

Our model uses fuzzy logic to describe the individual bird's drives (Lebar Bajec et al., 2005). These drives depend on a specific number of nearby neighbours, regardless of their distance (Cavagna and Giardina, 2008). Inter-bird occlusion is taken into account, i.e. nearby neighbours occlude those farther away (Kunz and Hemelrijk, 2012). In contrast to other models we have included social relations. We have included two types of animats solitary and social. The separation and alignment are the same for both types, but the cohesion drive is different. For social animats it models the desire to stay close to members of the same social group (e.g. family), for solitary animats it models the desire to stay close to nearby neighbours regardless of their affiliation.

To test the behaviour of the upgraded model we ran several simulations. The length of every simulation was 1800 frames (30 frames equals one second). The flock consisted of 20 animats. During the first simulation all 20 animats were solitary, so the flock was completely homogeneous. In our second simulation we had 15 solitary animats and 5 animats that belonged to the same social group. In our third simulation the flock consisted of 2 social groups with 5 animats in each and 10 solitary animats. Our last configuration had 3 social groups consisting of 5 animats each and 5 solitary animats. The flocks were left to roam freely inside a circular roosting area of 140 body lengths in diameter.

During our simulations we measured the *order* of the group (Vicsek and Zafeiris, 2012) and the number of flocks. We defined a *flock* as a group of animats that have influence on each other's behaviour. So to have two flocks there need



to be two groups of animats, in which no one from one group influences the behaviour of an animat in the other group, and vice versa. The order of the group is measured via the normalized velocity  $\varphi$ , which is calculated as

$$\varphi = \frac{1}{Nv_0} \left| \sum_{i=1}^N \vec{v}_i \right|. \quad (1)$$

## Results

As it can be seen from Figure 1, the behaviour of modelled animats is much more dynamic if the number of social groups is higher and the number of solitary animats is lower (more heterogeneity). The order of a homogeneous flock declines only when it reaches the edge of the roosting area and performs a U-turn. The behaviour is more diverse as well; splits and joins of flocks that consist of more than one social group are quite common and do not appear only at the boundaries of the roosting area.

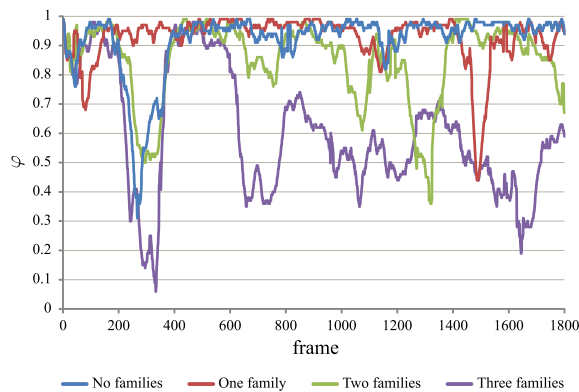


Figure 1: How the value of  $\varphi$  changes through time in differently structured flocks.

But the increase of the number of social groups and the decrease of solitary animats have a downside. A very low number of solitary animats produces separate flocks that fly independently and only seldom rejoin in a larger flock. Figure 2 shows the importance of solitary animats, as they are the main reason of re-joins of split flocks. So the most “natural” results were achieved when the number of solitary animats was the same as the number of social animats. Videos of simulations are available at <http://lrss.fri.uni-lj.si/cb/families.html>.

## Conclusion

Our simulations suggest that homogeneity might be an important factor for the lack of diversity in the displayed behaviour in computer models. Just with the addition of simple social relationships we managed to achieve complex manoeuvres in the form of splits and joins that resemble natural movements. What could be achieved with more complex heterogeneity is still to be researched.

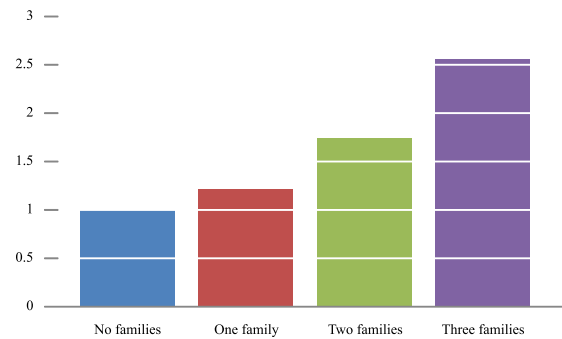


Figure 2: Average number of flocks during our simulations depending on structure of flock.

## Acknowledgements

We sincerely thank Frank H. Heppner and Maja Lebar Bajec for reading early drafts of the manuscript. This work was funded in part by the Slovenian Research Agency through the Pervasive Computing research programme (P2-0395).

## References

- Cavagna, A. and Giardina, I. (2008). The seventh starling. *Significance*, 5(2):62–66.
- Couzin, I. D., Krause, J., James, R., Ruxton, G. D., and Franks, N. R. (2002). Collective memory and spatial sorting in animal groups. *Journal of theoretical biology*, 218(1):1–11.
- Heppner, F. and Grenander, U. (1990). A stochastic nonlinear model for coordinated bird flocks. *The Ubiquity of chaos*, pages 233–238.
- Hildenbrandt, H., Carere, C., and Hemelrijk, C. (2010). Self-organized aerial displays of thousands of starlings: a model. *Behavioral Ecology*, 21(6):1349–1359.
- Jolles, J. W., King, A. J., Manica, A., and Thornton, A. (2013). Heterogeneous structure in mixed-species corvid flocks in flight. *Animal Behaviour*.
- Kunz, H. and Hemelrijk, C. (2012). Simulations of the social organization of large schools of fish whose perception is obstructed. *Applied Animal Behaviour Science*, 138:142–151.
- Lebar Bajec, I. and Heppner, F. (2009). Organized flight in birds. *Animal Behaviour*, 78(4):777–789.
- Lebar Bajec, I., Zimic, N., and Mraz, M. (2005). Simulating flocks on the wing: the fuzzy approach. *Journal of Theoretical Biology*, 233(2):199–220.
- Reynolds, C. (1987). Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, 21(4):25–34.
- Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., and Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226–1229.
- Vicsek, T. and Zafeiris, A. (2012). Collective motion. *Physics Reports*.

# Population Dynamics of Centipede Game using an Energy Based Evolutionary Algorithm

Pedro Mariano and Luís Correia

LabMAG – Dep. de Informática, Faculdade de Ciências, Universidade de Lisboa, Portugal  
plmariano@fc.ul.pt

## Abstract

In the context of Evolutionary Game Theory, we have developed an evolutionary algorithm without an explicit fitness function or selection function. Instead players obtain energy by playing games. Clonal reproduction subject to mutation occurs when a player's energy exceeds some threshold. To avoid exponential growth of the population there is a death event that depends on population size. By tweaking with the relation between payoff and energy and with death event, we create another dilemma that a population must overcome: extinction. We demonstrate this phenomena in the Centipede game. Simulations show that if players can only play one of the two positions of this asymmetric game extinctions are common. If players are versatile and can play both positions there are no extinctions.

## Introduction

Game Theory provides a series of tools to model how agents interact (Gintis, 2000; Shoham and Leyton-Brown, 2009; Fudenberg and Tirole, 1991). Evolutionary Game Theory (EGT) studies the population dynamics of individual agents (Maynard Smith, 1982; Hofbauer and Sigmund, 1998). Therefore it is natural to use games as a starting point in any study on interaction dynamics be it at the individual level or at the population level. Nash Equilibrium is one of the most important concepts in Game Theory (Nash, 1951). Its equivalent in population dynamics is Evolutionary Stable States. It posits that no player has incentive to move from its strategy because he would be worse. Since the publication of this result, several games have been proposed that have cooperative dilemmas: if players follow the equilibria they are worse off. Among these games we cite Prisoner's Dilemma (PD), Centipede, Ultimatum and Public Good Provision (PGP). When these games are repeated a known number of times, the theory, through backward induction (Fudenberg and Tirole, 1991; Shoham and Leyton-Brown, 2009), predicts the demise of cooperation. When players have limited resources such as memory, they may lack the ability to count how many stages have passed, so they cannot perform backward induction. In fact, earlier work on using finite state automata to play games has

shown the prevalence of cooperation (Neyman, 1985; Axelrod, 1997).

The capability of players finding the equilibria as recently been shown to be a hard problem. Research on the complexity of computing Nash Equilibria has established it as belonging to class PPAD or *polynomial parity argument, directed version* Papadimitriou (1994). This class sits between complexity classes FP and FNP which generalise classes P and NP from decision problems to functions computable in polynomial time. Examples of classes of games in this class include general sum games in normal form with two or more players belong to this class (Chen and Deng, 2006; Daskalakis et al., 2009), repeated games (Borgs et al., 2010) or stochastic games. These results show that if player wants to compute the equilibria he would spend an exponential time. Clearly biological system do not sit still that long. In fact their computational resources are limited and thus they have to use them in some efficient way. Early research on artificial systems used finite-state machines to model the behaviour of agents (Axelrod and Hamilton, 1981). This approach when applied to cooperative dilemma games resulted in agents playing the strategy profile that result in higher social payoff.

Instead of focusing on the (in)capability of computing the Nash Equilibria, one may turn into to the population dynamics. If the population is some evolutionary stable state this is not the result of players computing the best response to their peer in order to become dominant in the following generations. Instead it is a combination of population-level mechanisms such as mutation, selection and stochastic events.

Evidence from experimental play (Camerer, 2003) showed that people do not follow the equilibria and that instead cooperation is prevalent in many places. Therefore it may be counterproductive to discuss the equilibria of some system or the rationality of players (Aaronson, 2013). Instead, one should focus on the mechanisms behind the behaviour of real players and population and how they have overcome a series of dilemmas. Partner selection (Aktipis, 2004) or player connectivity (Santos and Pacheco, 2005) explain the prevalence of cooperation in these type of games.

We should be focusing on how evolution has overcome a series of dilemmas to produce the biodiversity that we can appreciate. One special dilemma is extinction. If evolution cannot surpass it then the model or system is not viable. Therefore it is important to consider a model that explicitly takes into account and allows it and does not have mechanisms to circumvent it by magic repopulation or by keeping a constant flow of energy resources (Yaeger, 2009).

In this paper we address the extinction dilemma. This event can appear in cooperative dilemmas because players choose the lower payoff profile. In order to introduce this event we have to create an Evolutionary Algorithm (EA) with varying population size. This has already been done in artificial ecosystems (Lenski et al., 2003; Ray, 1997) although the focus was not the study of how to avoid extinctions. The model that we propose is generic in that it can be applied to any game, just as the replicator equation (Taylor and Jonker, 1978), the Moran process or an Individual Based Model (IBM) (Grimm et al., 2006; McLane et al., 2011) can be applied to any game. This is done by simply interpreting a game as an energy transfer process and creating a player's life cycle that is driven by the energy acquired by playing games.

The rest of this paper is organised as follows. In the next section, we start by reviewing related work on models that can be used to study population dynamics while focusing on the ability to study extinctions. The following section is the major contribution of this paper as we describe the EA we have developed. The next section presents our characterisation of the Centipede game. Afterwards we present simulation results of our algorithm with the Centipede game. We finished the paper with a discussion section and a conclusions and future work section.

## Related Work

The study of evolutionary traits has used models based on differential equations such as the replicator equation (Taylor and Jonker, 1978; Nowak, 2006; Hofbauer and Sigmund, 1998; Maynard Smith, 1982; Gintis, 2000; Hofbauer and Sigmund, 2003) or the Moran process. They have been used to describe the broad behaviour of systems (Meadows et al., 1972).

There are a set of assumptions behind replicator equation (Roca et al., 2009). One assumes a considerably large or infinite population. Another assumes a well mixed-population such that everybody plays with everybody else. A similar approach is randomly pairing players. These are unrealistic assumptions and have led to alternative proposals. Among them are structured populations where players are placed in the nodes of some graph and interactions are restricted to links between nodes (Nowak et al., 1994; Szabó and Hauert, 2002). Despite not allowing varying population size, they have been used to model scenarios that may cause extinctions such as climate change (Santos et al., 2012).

Agent or IBM address the difficulties of creating a formal model of a complex system (Forrest and Jones, 1994). After a series of artificial ecosystems populated with these type of individuals (Ray, 1992; Lenski et al., 2003; Yaeger, 1994) specific protocols to construct such systems have emerged (Grimm et al., 2006).

There are IBMs that analyse the possibility of extinctions but they do that in specific contexts such as model population growth of endangered species (Beissinger and Westphal, 1998), tree mortality (Manusch et al., 2012), impact of logging activities in bird species (Thin et al., 2012). Some of these models are characterised by using specific differential equations or operate at higher level than the individual. Often they are specific to their case study and their methods are not directly transferable to another scenario.

McLane et al. (2011) provides a review of IBM used in the literature of ecology to address the issues of managing ecosystems. They presented a set of behaviours that individuals can choose in their life cycle: habitat selection, foraging, reproduction, and dispersal. In the papers that they reviewed, some used all the behaviours in the set while others used just one. Such set of behaviours could constitute the set of actions of some generic game played by animals. Moreover we can divide them in two sets, one where an animal obtains energy (foraging) and a second where an animal spends energy (habitat selection, reproduction, and dispersal).

While standard EGT models use either infinite populations or constant finite populations, IBMs have been used to model scenarios where populations could go extinct. This can happen because a player's actions do not provide him enough resources to reproduce. While their models are often used in specific problems it is important to create a general evolutionary algorithm that can be applied to any game and where extinctions can occur independently of game characteristics.

## Evolutionary Algorithm Description

The Energy Based Evolutionary Algorithm (EBEA) we have developed is characterised by a game. The concept of a game as an energy transfer is a redefinition of the payoff function. A game  $G$  is a tuple  $(N, A, E)$  where  $N$  is a set of  $n$  players,  $A = \{A_1, \dots, A_n\}$ , where  $A_i$  a set of actions for player  $i$ , and  $E = \{e_1, \dots, e_n\}$  is a set of payoff functions,  $e_i : A_1 \times \dots \times A_n \rightarrow \mathbb{R}$ , which we interpret as an energy flow. In this context, players are characterised by a strategy  $s$  and an energy level  $e$ . Each iteration of the algorithm has three phases:

**play** in this phase all players play the game and update their energy. Partners are selected randomly.

**reproduction** in this phase the players whose energy is above threshold  $e_R$  produce one offspring and their energy is decremented by this value. The offspring is mu-

tated with some probability. This is asexual reproduction with mutation.

**death** in this phase, the entire population goes through a carrying capacity event where the probability to die depends on population size. This contrasts with our previous work (Mariano and Correia, 2011) where the event mixed population size with player's age.

Regarding the relation between the payoff function and the energy function, we have extended our work in Mariano and Correia (2011). In order to compare the evolutionary dynamics of games with different payoff functions we scale the payoff  $\pi$  obtained by a player. This gives the following equation:

$$e \leftarrow e + \frac{\pi}{\bar{\pi} - \underline{\pi}}, \quad (1)$$

where  $\bar{\pi}$  and  $\underline{\pi}$  are the highest and lowest payoff obtainable in game  $G$  respectively, and  $e$  is a player's energy. The rationale for this equation is that player's energy increases proportionally to the payoff obtained in the game.

The chromosome codes the strategy to play the game  $G$ . The coding may range from direct, where actions are explicitly defined for all possible histories, to compact, where the strategy is represented in a symbolic form, for example by a set of rules to be executed by the phenotype. In both cases the result is either a specific action, for a pure strategy, or a probability distribution over the set of actions, for a mixed strategy. Notice that mutation operator must be adapted to the coding used.

Regarding the phenotype of a player, besides executing the strategy coded in the chromosome, it has associated an energy level. This is used for reproduction. As we have seen, an individual reproduces when his energy is above threshold  $e_R$ . This may also be considered as an indirect fitness assessment.

Unlike our previous work, in this paper we use random partner selection. Although this is an unrealistic assumption, we want to present a core evolutionary algorithm. From this core algorithm we can study what mechanisms are necessary to escape extinctions. In the context of PGP we have shown that partner selection can escape extinctions and promote cooperation for some parameter settings (Mariano and Correia, 2011).

The goal of this paper is to show what population dynamics we can observe with our evolutionary algorithm using Centipede as the game. In our current implementation of the algorithm, the random partner selection does not take into consideration if the game is symmetric or not. This means that in asymmetric games incorrect players may be matched. For instance, in the Ultimatum game two dictators may be paired. In this case both get zero energy.

In order to avoid exponential growth, in each iteration of the algorithm all players go through a death event. While in previous work we had a single event that mixed population

size with player's age, in this paper we only use population size. Death by old age is optional and is not strictly needed to avoid exponential growth. The probability of a player dying because of population size is:

$$P(\text{death population size}) = \frac{1}{1 + e^{6 \frac{K - |\mathcal{P}|}{K}}}, \quad (2)$$

where  $|\mathcal{P}|$  is the current population size and  $K$  is a parameter that we call carrying capacity. This probability is a sigmoid function. The exponent was chosen because the logistic curve outside the interval  $[-6, 6]$  is approximately either zero or one. In the advent of the entire population duplicating size, it will not go from a zero probability of dying to certain extinction.

From the description of our algorithm it is clear that there is no explicit fitness function nor selection function. Instead, the energy update function represented by equation (1) combined with the reproduction threshold induce a process where players that acquire more energy per game are able to reproduce faster (and spread their genes) than a player that gains less energy. A fit player is one whose strategy allows him to obtain more energy in the current population state.

The dynamics of this algorithm can be characterised by a Markov Chain. Supposing a well-mixed population, each state encodes a bag of pairs  $(s, e)$ , where a pair represents a player with strategy  $s$  and energy level  $e$ . Due to the carrying capacity we can impose a limit on the number of states, such that the probability of the population size passing this limit is negligible. This limit can be computed from equation (2). The Markov Chain has at least one absorbing state, namely the empty bag which corresponds to an extinction.

If there is a strategy profile of the underlying game that gives the minimum payoff to all players, then by equation (1) players using the corresponding strategies cannot increase their energy. This means that paths from states containing only strategies from this type of profile can only go to the empty bag state. However the time to walk this path may be large because death probability becomes negligible. From a practical standpoint the corresponding Markov states can be characterised as almost absorbing.

If there are no such strategy profile, then from any state other than the empty bag state there is a positive probability to reach any state in the Markov chain. This probability depends on event *death population size*, on event *mutation*, and on the *mutation operator*.

## Centipede Game

The Centipede game is a sequential game of perfect recall where in each stage a player decides if he keeps a higher share of a pot of money or decides to pass the pot to the other player (Rosenthal, 1981; McKelvey and Palfrey, 1992; Rand and Nowak, 2012). If the player keeps the higher share the



game stops. If he passes the pot is increased by some external entity. The game has some fixed number of stages. The payoff structure is constructed such that the payoff the deciding player obtains at stage  $t$  is higher than he obtains at stage  $t + 1$ . The game can be characterised by the initial size of the pot,  $p_0$ , how the pot is increased,  $p_i$ , and the pot share given to the player that decides to stop or not,  $p_s$ . We consider two methods to increase the pot: an arithmetic progression with difference  $d$ , represented by  $p_i = a(d)$ ; and a geometric progression with ratio  $r$ , represented by  $p_i = g(r)$ . The pot size at stage  $t$  is given by:

$$p(t) = \begin{cases} p_0 + d(t-1) & \text{if } p_i = a(d) \\ p_0 r^{t-1} & \text{if } p_i = g(r) \end{cases} \quad (3)$$

From this equation, the payoffs at stage  $t$  are  $\pi_D(t) = p(t)p_s$  for the player that stops and  $\pi_{-D}(t) = p(t)(1 - p_s)$  for the other player. The subscript  $D$  represents the player that decides to stop.

Figure 1a shows an example of the Centipede game for some parameter settings. Time goes from left to right. Since Centipede is an asymmetric game, we will consider two types of players: *first* represents the players that decide in odd stages; *second* represents the players that decide in even stages. In the extensive form game shown in figure 1a the types are represented by numbers one and two, respectively. In the last stage if player two decides to stop he receives the higher share of the pot. Otherwise the pot is increased but he receives the lower share.

In this paper we use a variant where the pot size is increased and then split as given by  $p_s$ . The parameters must obey the following set of conditions:

$$\begin{cases} p_s > 0.5 \\ d > 0 \wedge (p_0 + d)(1 - p_s) < p_0 p_s & \text{if } p_i = a(d) \\ r > 1 \wedge p_s > r(1 - p_s) & \text{if } p_i = g(r) \end{cases} \quad (4)$$

The second part in the second and third conditions represents the fact that  $\pi_D(t) > \pi_{-D}(t + 1)$

To decrease the number of parameters, we set the initial size of the pot to one,  $p_0 = 1$ . The admissible parameters of Centipede given by equation (4) can be represented graphically as shown in figure 1b.

The two methods to increment the pot create different pressures on players during our evolutionary algorithm. The difference in energy obtained per game is higher in the geometric method than the arithmetic variant. This means birth rate are different for both methods thus population viability is higher in the arithmetic variant.

The chromosome contains two genes. The first gene (binary) represents the player type while the second gene (natural number) represents the stage where it decides to stop the game. We will use  $t_\varepsilon$  to represent an action of the player that moves first and decides to stop the game at stage  $t_\varepsilon$ . Likewise, we will use  $t_s$  for the other player. Recall that in our

|       |                         |                                |
|-------|-------------------------|--------------------------------|
| $K$   | number of iterations    | $10^5$                         |
|       | carrying capacity       | {300, 400, 500, 600, 700, 800} |
| $e_R$ | mutation probability    | 0.1                            |
|       | reproduction threshold  | 20                             |
| $p_s$ | number of stages        | {4, 6, 8, 10, 12, 14}          |
|       | pot share               | {0.8, 0.9}                     |
| $p_i$ | arithmetic pot increase | {0.1, 0.2, 0.3}                |
|       | geometric pot increase  | {1.5, 2, 2.5}                  |

Table 1: Parameter values tested

current implementation of the algorithm, if players of the same type are paired, they obtain zero energy. Otherwise, they play the game.

## Experimental Analysis

### Simulation Settings

The purpose of the experiments is to study what kind of dynamical behaviour we can achieve with this new EA. Since the population size may vary, extinctions may happen because players cannot get enough energy to reproduce and they are slowly killed. We have tested different combinations of parameters. Regarding the parameters of the EA we varied only the carrying capacity in order to assess its impact on the occurrence of extinctions. We opted for using a single value for the mutation probability and reproduction threshold parameters. The number of iterations of the algorithm was  $10^5$ . Regarding the game parameters, we varied the number of stages in order to assess the amount of strategies on the population dynamics. We used high pot share to strengthen the backward induction argument. As for the pot increase method we opted to increase the pot share as given by equations (3) and used different differences or ratios. Table 1 shows the parameter values used in the experiments.

The initial population consisted in 10 players with chromosome type *first* and 10 players with chromosome type *second* both with the highest time to stop the game,  $t_\varepsilon$  and  $t_s$ . Whenever a player exceed the reproduction threshold and produce an offspring, he was subject to mutation with probability 10%. The mutation operator consisted in adding or subtracting to the time to stop the game,  $t_\varepsilon$  or  $t_s$ , a discrete Gaussian distribution with average zero and standard deviation one. The resulting value was constrained in the interval one to the number of stages in the game.

For each parameter combination we performed 10 runs in order to get some statistical information on the population dynamics. For each simulation run, per iteration, we recorded population size, number of births, number of deaths, number of players with each type and average time to stop for each type. If the population size dropped below two we stopped the simulation.

Figure 2 shows the plot of the probability of the event *death population size*, as given by equation (2) for the tested

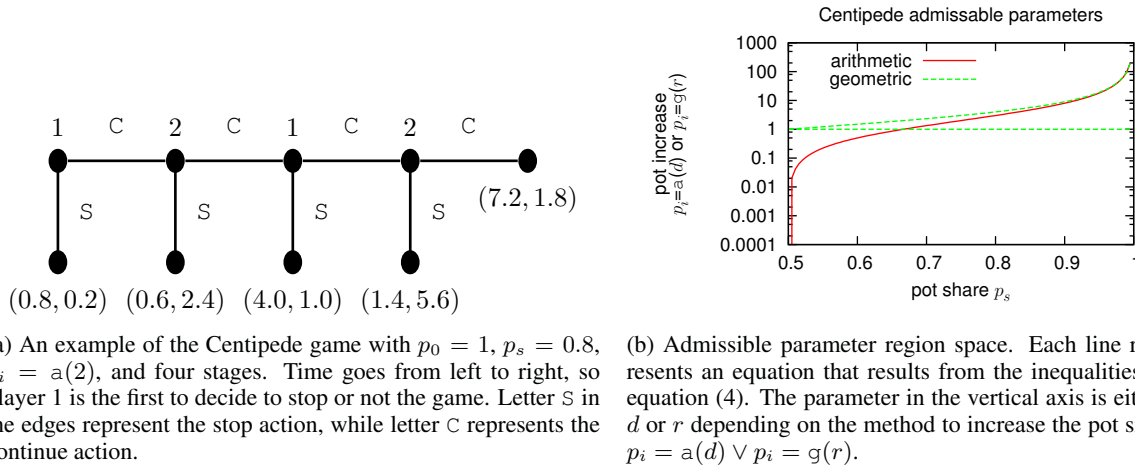
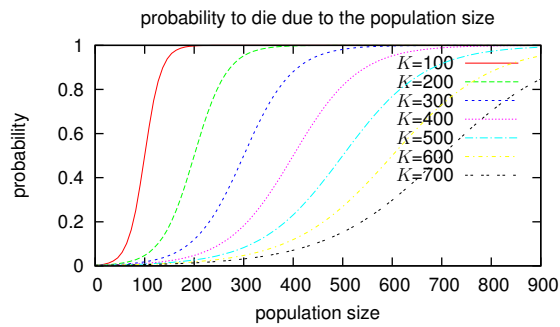


Figure 1: Schematics of the Centipede game used in the experiments.


 Figure 2: Plot of the probability of event *death population size* as given by equation (2) for different carrying capacity values.

carrying capacity values. As the parameter increases the plot becomes smoother meaning population size is unlikely to remain nearer the value of  $K$ . Moreover, as  $K$  increases, it becomes increasingly unlikely that the population doubles at specific iterations as more players had to synchronise their reproduction behaviours.

## Simulation Results

The major contribution of EBEA is the possibility of extinctions. In the evolutionary model that we have used extinctions can only occur if the population size drops below the number of players of the game. In the work that we report in this paper it is two. Another possibility is one of the player types extinguishing from the population. Regarding simulations using geometric pot increase, Figure 3 shows the number of simulations where some type of extinction occurred as a function of carrying capacity (vertical axis) and number of stages (horizontal axis). There is not a clear trend although a higher number of extinctions of second players compared to

first player extinctions is clear and extinctions of first players tend to increase with number of stages. Simulations with the arithmetic pot increase resulted in less extinctions for both types of players with around 45% less for second players and 95% less for first players. This is mainly due to a smaller difference (compared to the geometric increase) between stage payoffs.

We can also analyse what strategies are more common under this evolutionary algorithm. In the simulations that we performed the average time to stop the game,  $t_f$  and  $t_s$ , showed a decreasing trend to the smallest value, meaning players become less cooperative. Rand and Nowak (2012) argue that population size and selection strength affect the prevalence of cooperator strategies, meaning players that stop the game latter. In their paper they have used the geometric pot increase variant. However, their strategies could play both types. Given randomisation over partners, this means that on average a player is better never stopping the game. If we take the example of the Centipede game in figure 1a their players would get on average a payoff of  $(7.2 + 1.8)/2 = 4.5$ . When they decrease selection strength the weight of a player's payoff in his fitness decreases and selection becomes a random process. They only observe higher levels of cooperation when selection strength is mild, but we suspect that the cause is the ability of players playing both types. We have performed some simulations with our evolutionary algorithm where players could play both types. In these simulations we never observed an extinction. Regarding the geometric pot increase, we went from a scenario where some type of extinction occurred in 78% simulations to a scenario with no extinctions. Moreover players stopped the game at later stages. Switching from specialist players that only played one type to generalist players that played both types solved the problem of extinction.

This fact is very important given that, although our algorithm does not have an explicit fitness function, it has a high selection strength. If we examine the population dynamics of single simulations we often observe variations in population size that are correlated with player type's abundance and their strategies. When they stop the game at latter stages, they are able to acquire energy faster thus produce more offsprings per iteration. Therefore given two populations one that stops earlier and another that stops later and we run our algorithm without mutations, the second population outgrows the first.

## Discussion and Comments

The EA algorithm that we have presented does not have an explicit fitness function nor selection function. Instead players must acquire energy in order to be successful. This is an approach similar to individual-based models such as Echo (Forrest and Jones, 1994) or Avida (Lenski et al., 2003). Their models are ecosystems that use very specific games. Indeed one could formalise the interactions performed by the individuals in those systems as a game. With our approach we interpret games as an energy transfer process. In these systems there is also an energy concept. Individuals must perform some tasks in order to obtain energy tokens that are used to generate offsprings. While offsprings usually replace some stochastic chosen individual, in our approach they are added to the population. This is a similar approach to Yaeger (2009).

Scaling allows us to compare the evolutionary dynamics of games with different payoff functions: consider the number of offsprings per iteration. We could remove scaling, which means that energy range is equal to payoff range.

The EA has at least one absorbing state, namely the one corresponding to an empty population. A state of the corresponding Markov Chain may not be reachable if the population is filled with players that only get the lowest payoff. Therefore the only path is towards extinction, which can take a long time as we have seen in simulations using Centipede. This could be resolved if we had a death by old age event. This would put extra pressure on a population to escape extinction.

This process (energy dynamics and population control) is different from other approaches (Aktipis, 2004; Ray, 1997; Lenski et al., 2003). Even when they use energy, the focus is not the evolutionary algorithm and extinctions are not possible. Interactions between players are mediated by some game, which determines how much energy a player obtains. Therefore it is applicable to any game, either some simple game such as Iterated Prisoner's Dilemma (Aktipis, 2004) or a complex game where strategies are computer programs (Lenski et al., 2003). Population control is also independent of the game as only depends on population size.

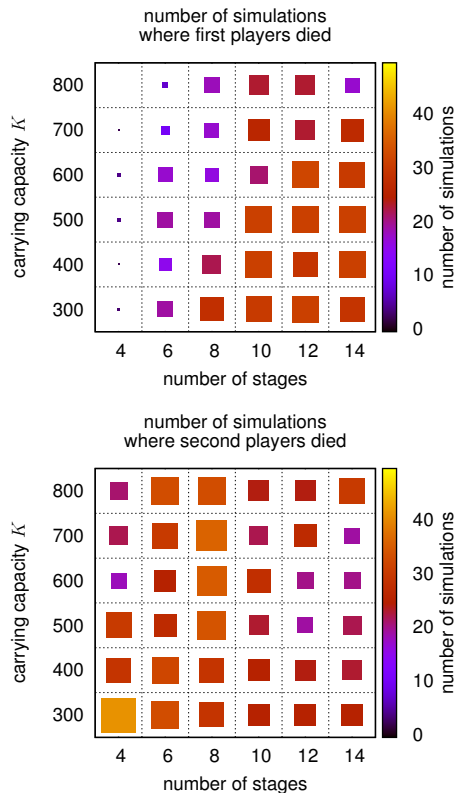


Figure 3: Results from the simulations with geometric pot increase. The plots show the number of simulations where first or second players got extinct – the higher the number of simulations the bigger is the square.

## Conclusions and Future Work

We presented an Evolutionary Algorithm (EA) that fits in the field of Evolutionary Game Theory (EGT) and of Individual Based Model (IBM). That is to say, players go through the stages of birth, growth, reproduction and death all mediated by some game that is characterised as an energy transfer process. A game instead of having a set of payoff functions, has a set of energy transfer functions. A player's chromosome codes his strategy and a player's phenotype executes the coded strategy and has associated an energy level. Reproduction occurs when a player reaches some energy threshold. In order to avoid exponential population growth, a death event, by population size, is performed in every iteration of the algorithm. This creates a new dilemma that players of a game must face, namely extinction.

Our EA can be compared to other models where energy managing is the focus (Lenski et al., 2003; Yaeger, 2009; Ray, 1997). While in these models extinction is circumvented or is not the focus, in our algorithm it is a dilemma and a danger that a population must keep evading. This is a harder dilemma than cooperation, because while with the latter a population may stay in some non-cooperative state for a long a time, an extinction is a dead end, it is the only absorbing state of our algorithm.

The algorithm is generic has it can be applied to any game, be it a simple one such as Centipede, Prisoner's Dilemma (PD), Ultimatum or PGP or a complex game whose action space is not explicitly given but instead there is some logical description to construct strategies to play the game.

We have applied our EA to the Centipede game. This is an asymmetric game with two types of players. Extinctions occur when players can only play one of the two types. This dilemma can be circumvented if players are able to play both types. Contrasting with Rand and Nowak (2012) we did not have to lower selection strength.

Concerning future work, we have some preliminary results regarding the application of our algorithm to other games such as Ultimatum, PGP or 2-player 2-action games. In the results obtained so far we have seen the occurrence of extinctions. We plan to increase the extinction pressure on players by adding a death by old age. This puts pressure on players to find mechanisms to avoid the fate of extinction. A possible mechanism is partner selection (Mariano and Correia, 2011). Another avenue of research is considering variable player's energy as is done in artificial ecosystems (Yaeger, 2009; Ray, 1997). This means that the fitness function can be interpreted as an energy transfer process not only from the environment to the player but backwards. This puts extra pressure in games that have costs associated to some actions such as PGP or some variants of 2-player 2-action games.

## References

- Aaronson, S. (2013). Why computational complexity theorists should care about philosophy. In Copeland, B. J., Posy, C. J., and Shagrir, O., editors, *Turing, Gödel, Church, and Beyond*. MIT Press.
- Aktipis, C. A. (2004). Know when to walk away: contingent movement and the evolution of cooperation. *Journal of Theoretical Biology*, 231:249–260.
- Axelrod, R., editor (1997). *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton Studies in Complexity. Princeton University Press.
- Axelrod, R. and Hamilton, W. D. (1981). The evolution of cooperation. *Science*, 211:1390–1396.
- Beissinger, S. R. and Westphal, M. I. (1998). On the use of demographic models of population viability in endangered species management. *The Journal of Wildlife Management*, 62(3):821–841.
- Borgs, C., Chayes, J., Immorlica, N., Kalai, A. T., Mirrokni, V., and Papadimitriou, C. (2010). The myth of the folk theorem. *Games and Economic Behavior*, 70(1):34 – 43.
- Camerer, C. (2003). *Behavioral Game Theory*. Princeton University Press.
- Chen, X. and Deng, X. (2006). Settling the complexity of two-player nash equilibrium. In *Foundations of Computer Science, 2006. FOCS '06. 47th Annual IEEE Symposium on*, pages 261–272.
- Daskalakis, C., Goldberg, P. W., and Papadimitriou, C. H. (2009). The Complexity of Computing a Nash Equilibrium. *SIAM Journal on Computing*, 39(1, SI):195–259. 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, MAY 05–23, 2006.
- Forrest, S. and Jones, T. (1994). Modeling complex adaptive systems with echo. In Stonier, R. J. and Yu, X. H., editors, *Complex Systems: Mechanism of Adaptation*, pages 3–21. IOS Press.
- Fudenberg, D. and Tirole, J. (1991). *Game Theory*. MIT Press.
- Gintis, H. (2000). *Game Theory Evolving - A problem-centered introduction to modeling strategic interaction*. Princeton University Press, first edition.
- Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S. K., Huse, G., Huth, A., Jepsen, J. U., Jørgensen, C., Mooij, W. M., Müller, B., Pe'er, G., Piou, C., Railsback, S. F., Robbins, A. M., Robbins, M. M., Rossmanith, E., Rüger, N., Strand, E., Souissi, S., Stillman, R. A., Vabø, R., Visser, U., and DeAngelis, D. L. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*, 198(1-2):115–126.
- Hofbauer, J. and Sigmund, K. (1998). *Evolutionary Games and Population Dynamics*. Cambridge University Press.
- Hofbauer, J. and Sigmund, K. (2003). Evolutionary game dynamics. *Bull. Amer. Math. Soc.*, 40(4):479–519.



- Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423(6936):139–144.
- Manusch, C., Bugmann, H., Heiri, C., and Wolf, A. (2012). Tree mortality in dynamic vegetation models - a key feature for accurately simulating forest properties. *Ecological Modelling*, 243(0):101–111.
- Mariano, P. and Correia, L. (2011). Evolution of partner selection. In Lenaerts, T., Giacobini, M., Bersini, H., Bourguine, P., Dorigo, M., and Doursat, R., editors, *Advances in Artificial Life, ECAL 2011: Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*, pages 487–494. MIT Press.
- Maynard Smith, J. (1982). *Evolution and the Theory of Games*. Cambridge University Press.
- McKelvey, R. D. and Palfrey, T. R. (1992). An experimental study of the centipede game. *Econometrica*, 60(4):803–836.
- McLane, A. J., Semeniuk, C., McDermid, G. J., and Marceau, D. J. (2011). The role of agent-based models in wildlife ecology and management. *Ecological Modelling*, 222(8):1544–1556.
- Meadows, D. H., Meadows, D. L., Randers, J., and III, W. W. B. (1972). *The Limits to Growth*. Signet.
- Nash, J. (1951). Non-cooperative games. *Annals of Mathematics*, 54(2):286–295.
- Neyman, A. (1985). Bounded complexity justifies cooperation in the finitely repeated prisoners' dilemma. *Economics Letters*, 19(3):227–229.
- Nowak, M. (2006). *Evolutionary Dynamics : Exploring the Equations of Life*. Belknap Press of Harvard University Press.
- Nowak, M. A., Bonhoeffer, S., and May, R. M. (1994). Spatial games and the maintenance of cooperation. *Proceedings of the National Academy of Sciences*, 91:4877–4881.
- Papadimitriou, C. H. (1994). On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532.
- Rand, D. G. and Nowak, M. A. (2012). Evolutionary dynamics in finite populations can explain the full range of cooperative behaviors observed in the centipede game. *Journal of Theoretical Biology*, 300(0):212 – 221.
- Ray, T. S. (1992). An approach to the synthesis of life. In Langton, C. G., Taylor, C., Doyne, J. D. F. J., and Rasmussen, S., editors, *Artificial Life II: Proceedings of the Second Conference on Artificial Life*, pages 371–408. Addison-Wesley.
- Ray, T. S. (1997). Evolving complexity. *Artificial Life and Robotics*, 1(1):21–26.
- Roca, C. P., Cuesta, J. A., and Sánchez, A. (2009). Effect of spatial structure on the evolution of cooperation. *Phys. Rev. E*, 80:046106.
- Rosenthal, R. W. (1981). Games of perfect information, predatory pricing and the chain-store paradox. *Journal of Economic Theory*, 25:92–100.
- Santos, F. C. and Pacheco, J. M. (2005). Scale-free networks provide a unifying framework for the emergence of cooperation. *Physical Review Letters*, 95(9):098104.
- Santos, F. C., Vasconcelos, V. V., Santos, M. D., Neves, P., and Pacheco, J. M. (2012). Evolutionary dynamics of climate change under collective-risk dilemmas. *Mathematical Models and Methods in Applied Sciences*, 22 (1)(1140004).
- Shoham, Y. and Leyton-Brown, K. (2009). *Multiagent Systems: Algorithmic, game-theoretic and logical foundations*. Cambridge University Press.
- Szabó, G. and Hauert, C. (2002). Phase transitions and volunteering in spatial public goods games. *Phys. Rev. Lett.*, 89:118101.
- Taylor, P. D. and Jonker, L. B. (1978). Evolutionarily stable strategies and game dynamics. *Mathematical Biosciences*, 40:145–156.
- Thinh, V. T., Jr., P. F. D., and Huyvaert, K. P. (2012). Effects of different logging schemes on bird communities in tropical forests: A simulation study. *Ecological Modelling*, 243(0):95–100.
- Yaeger, L. (1994). Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or PolyWorld: Life in a new context. In Langton, C. G., editor, *Proceedings of the Workshop on Artificial Life (ALIFE '92)*, volume 17 of *Sante Fe Institute Studies in the Sciences of Complexity*, pages 263–298, Reading, MA, USA. Addison-Wesley.
- Yaeger, L. S. (2009). How evolution guides complexity. *HFSP Journal*, 3(5):328–339.

## Cooperation of two different swarms controlled by BEECLUST algorithm

Tobias Meister, Ronald Thenius, Daniela Kengyel and Thomas Schmickl

Artificial Life Laboratory, University of Graz, Austria.  
ronald.thenius@uni-graz.at

### Abstract

In this work we investigate how two autonomous agent swarms, controlled by the BEECLUST algorithm are able to cooperate. The task is to locate two different target areas, which are located near each other. Therefore we developed an individual-based NetLogo model to simulate two different agent swarms moving in a temperature gradient. Both agent swarms are controlled by the BEECLUST algorithm, which is inspired by honeybee behavior. We found out that the two cooperating agent swarms are able to locate the target areas, independent of the ratio of the agents.

### Introduction

An eusocial insect colony, which consist of large numbers of individuals can be interpreted as a “superorganism” (Oster and Wilson, 1979). This superorganism is able to solve more complex tasks than the single individual members of the superorganism. The thermal reaction of worker honeybees (Ohtani, 1992), which leads to thermal homeostasis in the honeybee colony, is an example for the self-organisation and the swarm-intelligent behavior (Millonas, 1992) of social insects and one main inspiration for the experiments described in this paper. Several studies (Heran, 1952; Ohtani, 1992) have shown that young honeybees prefer to aggregate in, or near an area with a surrounding temperature of 36°C. The bees move randomly and form longer lasting clusters in warmer, than in cooler zones. We use this simple, but efficient behaviour of honeybees for our experiments. The algorithm derived from this behaviour is called BEECLUST (Bodi et al., 2012, 2011; Schmickl and Hamann, 2011). It is based on the following simple rules:

- (1) The agents move randomly through the arena. Whenever an agent detects an obstacle, it stops and checks whether the obstacle is another agent or a wall.
- (2) If the obstacle is a wall, the agent turns randomly and continues with step 1.
- (3) If the obstacle is another agent, the agent measures the local temperature and calculates his individual waiting time, dependent on the local temperature, according to a

sigmoidal function.

- (4) After the waiting time is over, the agent continues with step 1.

These simple actions of individual agents lead to resource-saving behaviour. Recent works (Kengyel et al., 2011; Kernbach et al., 2009) have shown that artificial agents controlled by this algorithm react flexible regarding environmental changes. In the work at hand we investigate how honey bees age polyethism influences this system. Age polyethism means that in a honeybee colony individuals of the same age perform the same task, and that a given task is often associated with a given age. Examples for such tasks are, collecting nectar in the environment, brood care and the cleaning of the honeycombs. The location of these tasks are not always spatially separated, but can be located near each other, or even within the same area, e.g., broodcare and wax manipulation. It was shown by Bodi et al. (2012) that agents, controlled by BEECLUST, that have identical sensors, but differ regarding their temperature optimum, are able to cooperate well in a complex environment. The question we raise here is, how good can two different groups of agents, that have different sensors (and therefore different tasks) cooperate, if the tasks are located near each other. Based on the results of Bodi et al. (2012), who described the negative influence of jamming effects on groups of agents operating in the same area, and the positive affects of cooperation subgroups we formulated the following hypothesis: Two non-identical agent swarms controlled by BEECLUST are able to cooperate for a given ratio of the agents in the two groups.

### Materials and Methods

To answer this question we performed and simulated our experiments in NetLogo (Wilensky, 1999). The simulated area has a size of 16x16 patches. We implemented two different task-areas with a distance of 5 patches and two different agent-swarms *A* and *B*, acting parallel in the same environment. Both swarms had the same properties and comply with the rules of the BEECLUST algorithm. The two differ-

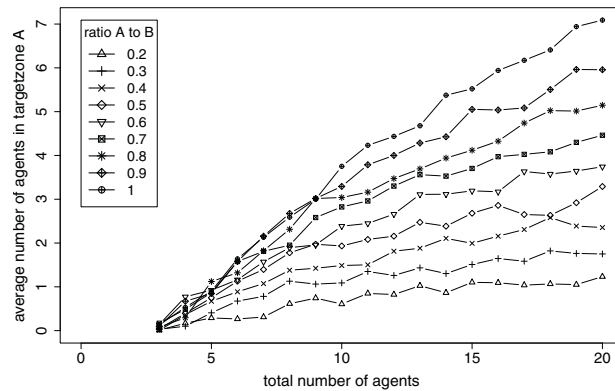


Figure 1: Absolute number of  $A$  agents, aggregated in  $T_A$ ;  $n = 100$ .

ent task-areas  $T_A$  and  $T_B$  were implemented as gradients in the environment, scaling from a value of 1 in the maximum to 0 in the environment. The size of  $T_A$  was the quarter of the size of  $T_B$  to simulate a highly specialised task near an area of a more general task.

As mentioned above the length of the waiting time of a single agent is determined by the local value of the gradient. The sigmoidal waiting-curve was identical for both agent-swarms  $A$  and  $B$ . The maximum waiting time for both swarms was 1000 timesteps.

We observed and analysed the percentage of agents of  $A$  aggregated at  $T_A$ . The tested population size, including  $A$  and  $B$ , ranged from 3 to 20 individuals. The ratio of  $A$  to  $B$  was varied from 0.2 to 1, rounding was always done towards the next bigger number of  $A$ . Each experiment ran for 3600 timesteps and was repeated 100 times.

## Results and Discussion

It showed, that, by increasing the total amount of agents, the average number of  $A$  increased linearly in the target area  $T_A$  (see figure 1). Surprisingly it further showed, that, in contrast to our hypothesis (mentioned above), the relative amount of  $A$  in the target zone was highly stable against changes of the ratio of  $A$  to  $B$  (see figure 2). This means, that even a single agent can operate within a group of agents with another task (or even another sensory system) without any loss of efficiency. Due to this we can falsify our hypothesis, that the cooperation of two BEECLUST controlled groups of agents is depending on the ratio of these two groups. To which degree these results can be used in the field of swarm robotics to develop selforganised heterogeneous robot swarms based on BEECLUST is the topic of ongoing research at the moment.

## Acknowledgements

This work was supported by: EU-IST-FET project SYMBRION, no. 216342; EU-ICT project REPLICATOR, no.

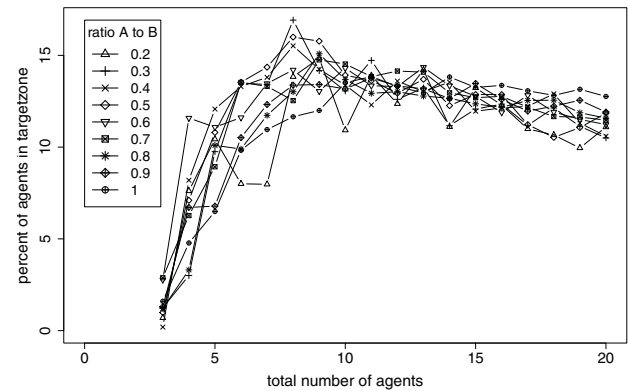


Figure 2: Average percentage of  $A$  agents, aggregated in  $T_A$ ;  $n = 100$ .

216240; EU-ICT ‘CoCoRo’, no. 270382; EU-ICT ‘ASSIS-Ibf’, no. 601074; Austrian Science Fund (FWF) research grants: P19478-B16 and P23943-N13 (REBODIMENT)

## References

- Bodi, M., Thenius, R., Schmickl, T., and Crailsheim, K. (2011). How two cooperating robot swarms are affected by two conflictive aggregation spots. In *Advances in Artificial Life. Darwin Meets von Neumann*, pages 367–374. Springer.
- Bodi, M., Thenius, R., Szopek, M., Schmickl, T., and Crailsheim, K. (2012). Interaction of robot swarms using the honeybee-inspired control algorithm beelust. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):87–100.
- Heran, H. (1952). Untersuchungen über den temperatursinn der honigbiene (*apis mellifica*) unter besonderer berücksichtigung der wahrnehmung strahlender wärme. *Zeitschrift für vergleichende Physiologie*, 34(2):179–206.
- Kengyel, D., Schmickl, T., Hamann, H., Thenius, R., and Crailsheim, K. (2011). Embodiment of honeybee’s thermotaxis in a mobile robot swarm. In *10th European Conference on Artificial Life (ECAL’09)*, volume 5777/5778 of *LNCS*. Springer.
- Kernbach, S., Thenius, R., Kornienko, O., and Schmickl, T. (2009). Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic swarm. *Adaptive Behavior*, 17:237–259.
- Millonas, M. M. (1992). Swarms, phase transitions, and collective intelligence. Technical report, Los Alamos National Lab., NM (United States).
- Ohtani, T. (1992). Spatial distribution and age-specific thermal reaction of worker honeybees. *Humans and Nature*, 1:11–26.
- Oster, G. F. and Wilson, E. O. (1979). *Caste and ecology in the social insects*. Princeton University Press.
- Schmickl, T. and Hamann, H. (2011). BEECLUST: A swarm algorithm derived from honeybees. In Xiao, Y., editor, *Bio-inspired Computing and Communication Networks*. CRC Press.
- Wilensky, U. (1999). Netlogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.

## The perception of potential: interference, dimensionality and knowledge

Jonathan Impett<sup>1</sup> and Leonardo Impett<sup>2</sup>

<sup>1</sup>School of Media and Performing Arts, Middlesex University, London

<sup>2</sup>St John's College, Cambridge

j.impett@mdx.ac.uk

### Abstract

This paper presents a system that investigates the sonification of wave interaction in a performance space and its interaction with a live performer – the illumination of sonic activity within a real space, in contrast to conventional ALife algorithmic, event- or material-based approaches. The model maintains three parallel representations of the entire live/virtual system: wavespace, symbol space and performance space. The cross-modal analysis and representation of behavior is important to the evolution of the system, which displays emergence on multiple levels of structure. Micro-evolution takes place within the population of wave-emitting and –listening agents. A higher level of structure emerges from their aggregate in interaction with the live performer, and a formal level as symbol space learns from the performer. Cross-modal representation is seen as a significant factor in the evolution of Western art music, in the development of multi-leveled structure and of work that affords many dimensions of engagement. We discuss the nature of knowledge produced through working with such systems and the role of the subject in ALife-generated knowledge. New models of simulation-derived knowledge are seen as important to cultural understanding.

### Introduction

#### ALife and creative music systems

The relevance of the ALife paradigm to music is well established. Dynamic, evolving populations of behaviors interacting within a constrained space, the emergence of structure unforeseeable from initial conditions, the role of self-simulation and learning and the negotiation of otherwise intractable networks of relationships are system properties that resonate with musical minds.

ALife approaches are particularly attractive to artists working in real-time or interactive environments (see, for example, Miranda 2011, Miranda and Biles 2007). Stricter algorithmic systems are locked within their fixed aesthetic and behavioral dimensionality. ALife architectures appear to be a way of distributing creativity (you can't make 44,100 good decisions a second); composer, performer or environment effectively become co-agents or super-agents. At the same time, there is often a suggestion that such systems in turn might reveal something of the mysteries of music, or of human aesthetic responses, or of sound-based evolution. In the general case, we are dealing with a constructive metaphor; any

verisimilitude is aesthetic. This is not to discount the knowledge inherent in such experience. Indeed, we shall discuss below how the example of an ALife music system can usefully raise questions about the relationship of the subject with such models and the nature of knowledge thus produced.

Interactive music applications have been the most fertile area for development. Technology affords exploration of the space between conventional notions of improvisation and composition, a space non-navigable by traditional means. Concepts such as real-time composition present non-trivial questions as to the relationship between apparently opposed activities of instantaneous decision-making and reflective architectural planning – apparently, because the instant (James's *specious present*) is crucially informed by intention, expectation and habit, and formal architecture is invariably modulated by the sequential, situated development of initial conditions.

#### The present system

ALife-based music systems tend to work on an event basis. That is, they deal with notes of sound objects or MIDI data. Such decisions are pragmatic (this devolves much of the processing), perceptual (we tend to think we listen to music in terms of notes) and cultural (the study of Western music remains largely text- or symbol-based). Artists such as Ryoji Ikeda and Carsten Nicolai work with the evolution and interference of wave-forms, but as entities abstracted from their environment (Ikeda and Nicolai, 2011). Feedback-based work exploits sonic characteristics of particular spaces and technologies (Alvin Lucier's *I am sitting in a room* and Nicolas Collins' *Pea Soup* are canonical examples). This system discussed here attempts to address the apparent dichotomy between “compositional” approaches based on the behavior of pre-designed materials and “environmental” approaches that explore a context. These views have distinct discourses; in the current UK Research Excellence Framework “Sound Art” falls within the purview of fine art, “Composition” under performing arts.

The present system sets out from a different perspective. We consider the real space of performance as a unitary volume within which micro-sonic wave activity is ubiquitous. The apparently silent blank canvas of music is illusory, such an anechoic reality even distressing. Such a view echoes Ingold's observations on soundscape:



... neither sound nor light, strictly speaking, can be an object of our perception. Sound is not what we hear, any more than light is what we see. The scaping of things - that is, their surface conformation - is revealed to us thanks to their illumination. When we look around on a fine day, we see a landscape bathed in sunlight, not a lightscape. Likewise, listening to our surroundings, we do not hear a soundscape. For sound, I would argue, is not the object but the medium of our perception. It is what we hear in. (Ingold 2007, 11)

In his early work on sound, Bill Viola proposed a taxonomy of sonic behaviors, likewise developed from an analogy with light:

A partial list of some of the most basic physical phenomena studied by the acousticians reads like a set of mystical visions of nature:

Refraction ...  
 Diffraction ...  
 Reflection ...  
 Interference ...  
 Resonance ...  
 Sympathetic Vibration ...

Each of these phenomena evokes wonder, even after their scientific representations have been rationally understood. [...] The processes of contemporary media systems are latent in the laws of nature – they have existed in various forms since the beginning of history. (Viola 2013, 41-2).



Figure 1: interference phenomena exhibited by the present wavespace system

The work presented here was developed for one of a sequence of compositions exploring wave phenomena in situated sound. It investigates the use of interference phenomena (fig. 1); others explore refraction and diffraction. Reflection plays a role in all three, as a key component in defining the relationship between space and activity.

We take a wave-based approach to modeling activity within the space. The potential complexity of even aurally trivial musical constructs presents a challenge. Instead of attempting to consider an intractable number of relationships we approach this complexity by looking at the behavior of the gradients and interference patterns generated by the musical material-generating agents that inhabit the space. In this respect, meteorology perhaps provides an analogy; a tornado is not an external event projected into a neutral space, but a product of dynamics within the environment itself. Nevertheless, in the domain of human activity it is best understood and responded to as an autonomous entity with its

own behavior. We might think of the process presented here as second-order sonification.

We suggest that an important component of musical composition is the remapping of materials and phenomena between different modes of representation – from aural (the sonic imagination) to symbolic (notation) to physical (working at the piano), for example. The present system maintains multiple views of its space: representations of its “real”, virtual/mathematical and symbolic or eventual state.

## Technical Description

### Implementation

In this implementation the performer uses a metatrumpet – a microcontroller-extended instrument that communicates its internal and external soundworlds and all physical activity to the computer via Bluetooth and radio. The system software is written in C++, using Max/MSP as interface and sound engine; they communicate using the UDP-based OSC protocol.

### Wave space and performance space – a double bind

The system architecture incorporates three parallel spaces: a virtual wavespace based on a graphical model (maintained in the C++ programme), a symbolic space using CMMR-like representations (quantized to semitones and in time) and the performance space where sounds are played and heard in the physical world (both handled in Max/MSP) (fig. 2).

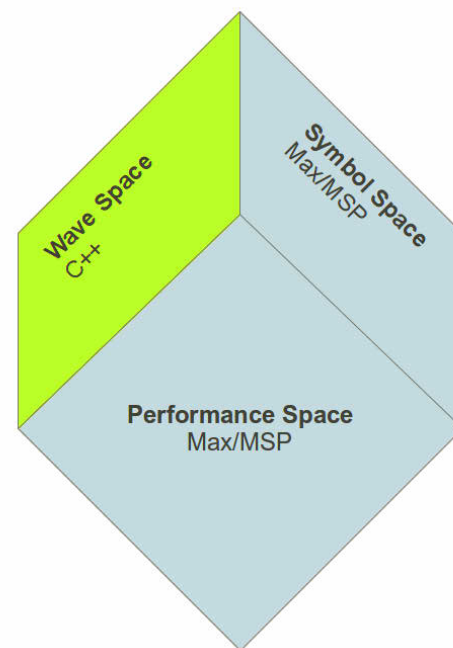


Figure 2: The three parallel spaces can be metaphorised geometrically as three orthogonal planes; navigation in one plane does not affect navigation in the other two, but all three are affected by the dynamics of the system

The performer initiates wave-generating agents. These are singularities in the wave-field - effectively beacons radiating sine waves. The wave behavior of each agent is initialized with values of initial position, frequency, speed of sound and spatial attenuation. Associating speed and attenuation with each individual agent allows for distortions of the space; different agents might find themselves in media through which sound propagates at different speeds. This also permits the effective representation of nonlinear spaces much larger than the actual performance space - large but bounded acoustic spaces within which interference artifacts fall within the range of human hearing. The agent's position is recalculated at every time step, and can be fixed, periodical, rule-based or learned (fig. 3). A repulsive force of proximity prevents terminal convergence. Once initialized, an agent generates one or more sine waves for the duration of its existence. In the initial model, agents are active or not. For reasons of practicality, the model is a steady state approximation. The field is conservative; it has no memory. If an agent is destroyed, all its effects immediately disappear from the space.

```
struct wave{
    float amplitude; //The starting amplitude of the emitted
    wave
    float xpos;      //The x,y positions of the agent
    float ypos;      //...
    float pointvelocity; //The agent's velocity through space
    float omega;     //The emitted wave's angular velocity
    float beta;      //The emitted wave's spacial attenuation
    float learn;     //The agent's learning aggressiveness
    float contr;     //The agent's estimated contribution...
    float control;   //...and in the previous timestep, for
    comparison
    float angle;     //The agent's spatial direction
    bool dominant;   //The agent's state: is it in the most
    harmonic pair?
};
```

Figure 3: C++ data struct for each wave-emitting/-listening agent

At each system time step we know the phase dependent contribution of each agent to the resultant signal at each point in space. In the initial model, system speed is 1 KHz, and the dimensions of the wavespace 256 \* 256 points. Interference patterns within a space are perceived clearly when the dimensions of that space are an order of magnitude above the wavelengths in question. If we assume the scale of a typical performance space to be of the order of 10m, this grid allows for the representation of signals within the range of human hearing - wavelengths of the order of 1m. A balance of precision and computability is particularly critical in a real-time system servicing streams of audio output.

### Sound production

The total wavespace field is calculated by summing the contribution from each agent at each point. Identifying areas

of interest - loci of active interference and potentially emergent behavior - requires deconstructing intuitive understandings. The wavespace itself is indifferent. Initial strategies included looking for peaks in the image derivative - effectively high pass filtering - and edge detection. Both produced indiscriminate amounts of data and proved computationally expensive. The strategy adopted is to identify points of maximum interference by looking at the difference between the scalar sum and the vector sum of the contributing waveforms.

Figure 4 shows the gradients of interference produced by three agents in such a space, and the point of perceived maximum interference calculated as described above.

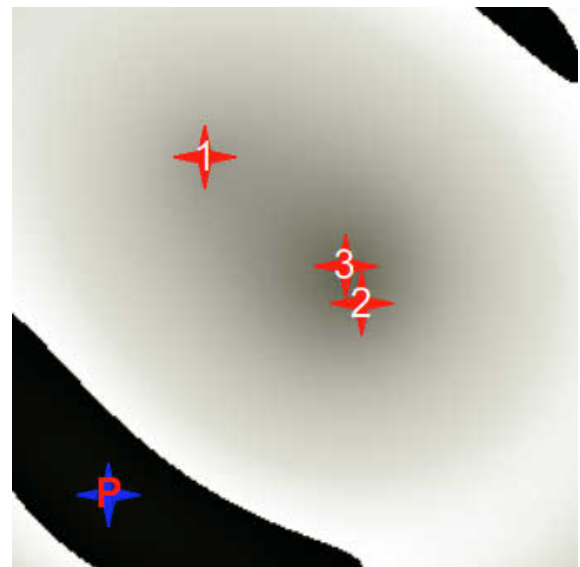


Figure 4: Three agents in wavespace (1-3) and the point of maximum perceived interference (P)

Interference in this sense is result of phase differences between sinusoids. Another less formal understanding of interference might entail the production of additional noise or artifacts. This is an embodied phenomenon. In the case of hearing, for example, modulation artifacts are a function of the response of the basilar membrane (Moore 2002).

Having localized the point of maximum interference, a hypothetical sound is produced by taking the weighting the various agent audio signals by values stored in a matrix that represent their estimation of their contribution at that point. This data is valid for one wavespace timestep (1ms). The audio stream is subsequently filtered at timestep frequency to mitigate windowing artifacts.

We consider this signal as being emitted radially by the maximum as a point source in wavespace. This sound is never heard directly. In keeping with our embodied understanding of interference, the beacon agents also become listeners and performers in performance space. We then hear the combination of their listenings to the combined reflected

signal, which now themselves exhibit interference due to their phase displacements. The agents are dynamically located in physical space using IRCAM's *Spat* spatializing application within Max/MSP (<http://support.ircam.fr/docs/spat/3.0/spat-3-intro/co/overview.html>).

This information is sent via OSC to the agent's Max/MSP avatar. The audio stream of each agent is constructed by additive synthesis in Max/MSP from the discrete frequency spectrum. It is then buffered to maintain a stable audio stream. It is delayed and attenuated according to its position in wavespace and its local speed of sound. In addition to the frequency/amplitude/phase data representing the point source signal, the performance space is informed of the positions of the other agents and of the maximum. Before synthesis, the agent can make certain musical decisions. In particular, frequency addition artifacts above the range of human hearing (c. 20 KHz) are transposed down to become audible. This initiates a recursive exchange between wavespace and performance space through which increasingly complex sounds can accumulate.

### Accumulating agency

In an early iteration of the system, agent position is determined dynamically by a fixed velocity and a direction of motivation towards the point of maximum interference. A dance develops as agents track the maximum point-source, the location of which is in turn determined by the agents' position. Quasi-oscillation, temporary attractors, reflections, line-following, convergence/divergence, and leaps through space all occur. The latter is not just a function of the bounded wavespace. The real performance space it reflects is also bounded, and such events are therefore considered to be of potential musical-structural significance. Figure 5 shows the movements of three agents and the point of maximum perceived interference for the 56 timesteps leading to the positions shown in Fig. 4.



Figure 5: movement of three agents and point of maximum perceived interference through 56 timesteps preceeding fig. 4

**Microevolution.** A more advanced system is developed where a dynamic number of agents  $N$  navigate the space, and the system judges harmonic relationships between pairs of agents in an  $N \times N$  matrix. The position of the smallest element in this matrix indicates the pair of agents with the strongest harmonic relationship. In a reflection of the bio-evolutionary paradigm, the agents which are not part of this pair reproduce imperfectly and are destroyed, creating imperfect re-initializations of themselves on destruction. Consistent with the spatial model, the degree of mutation is dependent on the proximity of the agent to the point of maximum interference, representing information loss. After successive generations, a different set of two agents will exhibit a stronger relationship, and the set of agents which reproduce will change accordingly. Each agent continues to have a behavior dictated by both the initializing conditions outlined above, and learned behavior (controlling agents' own frequency and direction of spatial movement) maximizing their contribution to the point of maximum interference. The aggressiveness of the learning algorithm is itself an initialized parameter. Agents within the harmonic relationship are understood as changing their behavior whilst agents outside the harmonic relationship change their nature.

**Polyphonic rhythm.** The performed sounds of each agent are subject to a further thresholding process. Agents calculate their own contribution to the sound perceived at the maximum point source by comparing the interference intensity at that point with and without their contribution.

They become active as performers only when this contribution exceeds a threshold – that is, the sound of each agent is turned on and off with a rhythm of variable linearity with windows of periodicity. This creates a polyrhythmic fabric, each strand of which is related to the virtual maximum. The texture can be inverted – transformed to its negative by reversing the threshold test. A challenge in music systems with any degree of automation is to generate the multiple levels of structure found in art music. The system again analyses the relationships between individual agents as pairs, but this time looking for convergence in audial performance space. When both audio streams of a pair of agents converge, giving out a 'drone', a timer is initialized; after a fixed time period, both agents are silenced in performance space. However, they continue to operate in both the wave and symbol spaces, which under the micro- and macro-evolutionary paradigms outlined eventually cause the pair of agents to exhibit different audial-spatial behavior, and their inaudible audio streams diverge. Under these conditions, the audio streams of the two agents are restored to performance space, again becoming audible to the performer (fig. 6).



Figure 6: Thresholding behavior of three agents over time

## Macroevolution – symbolic representation

A third parallel space maintains symbolic representations of performance activity. Streams of sound output, synthesized and live, are analyzed in terms of pitch and rhythm. In both domains this constitutes a process of re-representation and categorization roughly analogous to human cognitive behavior; the point here is not anthro-verisimilitude but process architecture. Perceived pitch is quantized to semitones, rhythm to a 50 ms. grid. Inevitably these processes lead to loss of precision, of information, but they allow for the apprehension of relationships on other levels. The aggregate wavespace output is then analyzed in terms of harmonicity, density of behavior (a measure of number of events perceived over time), “noisiness” (signal-derived value) and stability of amplitude. The rhythmic profile of individual streams are compared by searching for a lowest common denominator – an additive pulse – and looking for matches or complementarity.

Initial threshold values are set for these parameters together with a global coefficient of surprise. Principal Component Analysis is used to produce values representing the balance of factors in the current state. Symbol space directs change in wavespace under two kinds of condition:

- when a subset of parameter thresholds is exceeded
- when the performer triggers a change. In this case, symbol space learns the current PCA state. When the same PCA state recurs (within a given tolerance), symbol space will subsequently trigger the same changes autonomously. PCA thus represents a higher-level representation, less dependent on specific detail of behavior. In this way a repertoire of formally salient states evolves through interaction with the performer.

The changes sent from symbol space to wavespace are distributed among agents by the main loop – effectively a super-agent. They relate to the following properties:

- number of agents: the population may be forced to multiply or divide
- frequency bands: the potential frequency space inhabited by each agent (initially 20-20,000 Hz) may be divided into separate bands, such that the agents are distributed among bands of varying width and distance.
- range and distribution of speeds of sound
- triggering threshold
- repertoire of wavetables (wavetables written from live sound and from wavespace output are added to the initial sine tables)

The system thus evolves over longer time-scales, approximating to formal time in compositional terms, by learning a repertoire of transition states through interaction with the performer.

## The performed space

The live performer is incorporated through performance space. FFT data is passed to wavespace where the performer appears as an agent among others, contributing to the aggregate interference pattern. This is captured by a

microphone at the position of the performer, such that the performed sound of the agents as heard by the performer also figures in this stream. Clearly the performer is not as free to move in wavespace (and doesn’t have the mental capacity to calculate interference maxima). Instead, motion data from the metatrumpet directs both movement in virtual space and the positioning of sound in performance space. The performer controls the range of frequency components to be passed. In a four-band implementation, for example, one might select components 9-12 to avoid the perceived pitch and its low integer multiples, focusing the system’s attention on the activity of higher partials. To facilitate calculation and avoid additional artifacts (from the arbitrary relationship of table length to frequency), a cycle at the lowest perceived frequency is written into a wavetable.

The performer can work with current tendencies, attempting to counter or encourage them, trigger new agents directly, or mark a current state as a moment of transition in symbol space.

## Further Developments

Our intention is to enhance the relationship between virtual wavespace and actual performance space by incorporating an acoustic space-tracing algorithm (Dokmaniç et al, 2013). This impulse response-based technique can be incorporated both in musical terms, by allowing the perceived interference patterns to evolve from a single impulse, and technologically, in that the necessary microphones already form part of the system. Wavespace precision will be unchanged, but the representation can be mapped nonlinearly onto the physical space.

Agents will be empowered to search their own perceived spectrum, to select and transpose series of higher partials that contribute a greater range of color to the whole.

The performer will be able to mark a state as stable as well as transitional. If the potential for any of the evolving repertoire of stable states is perceived in symbol space, wavespace will be encouraged in this direction.

A predictive function will be incorporated in symbol space, in the hope of enhancing the measure of novelty. It will learn by correlation with the performer’s real-time decisions.

## Conclusion

The re-representing of behavior across modal boundaries is characteristic of Western art music of many kinds (Impett, 2009). Indeed, we could understand its historical development as being the incremental and technological exploration of this possibility. Polyphony – a complex system which cannot be fully resolved or reduced (unlike, say a simple melody-and-bass structure) – is the most characteristic property of the last thousand years of art music. The kind of cross-modal interaction explored here allows for the emergence of potential structure on multiple levels and their interaction to generate a complex, compound architecture in which detail on one level relates to form on another. We attempt to create a dimensionality of engagement (of expectation, of analysis, of interpretation) higher than that of purely algorithmic or performer-driven systems. Looking at interference and convergence provides a way of grasping the state of a



complex system and deriving understanding of its potential architecture.

This system can be tuned to the tastes of the musician; it is replicable but not falsifiable. Rather than invalidating aesthetic models as knowledge-producing environments, this obliges us to confront the role of the subject in knowledge production. The apprehending subject is part of the knowledge producing system; production and apprehension become inseparable. A-Life models are used to model societies, cultures, populations and complex systems, but rarely to understand the individual. This may be partly historical, in contradistinction to GOFAI for which the implicit paradigm was perhaps a cartoon of human intelligence. We cannot avoid the role of the subject, however, particularly as such open systems present behavior that cannot be represented in its entirety. In exploring the nature of this new kind of knowledge, Delanda describes their behaviors as singularities in a *space of potential* (Delanda 2011, 18).

This work is part of a wider theoretical project, the *work-without-content*. It takes its cue from Giorgio Agamben's observation that in a culture of infinite difference, of the right to expression in unique individual languages, the common experience on which cultural exchange is possible becomes the confronting of the blank page. He describes the instantiation of such work as giving form to potentiality (Agamben 1999). Here the material derives entirely from the relationship of performer and space.

What is the relationship between the subject as pilot of the system and subject as co-agent? The situation can be characterized in terms of power and rules; it is effectively a legal question. We can look to Agamben once more. He considers the role of the sovereign, the individual in relation to which laws obtain, and yet who may stand outside them. He can be assassinated, but not condemned to death. Likewise his *homo sacer*, the outlawed subject whom the state will not kill but another individual may choose to (Agamben 1998). Sovereign power or *homo sacer*, the rules and behavior of a system such as this produces knowledge – if the production of truths is how we understand the function of art (Badiou 2001, 41-57) – only by incorporating such a subject.

There is another important aspect of emergent knowledge-production - an idea that science, in turn, might take from critical theory. Such knowledge is experiential and situated, it denies commodification and resists attempts at repetition and falsification. In this respect, music might be seen to provide a model for the new knowledge. Meanwhile the commodification of culture proceeds unabated. Science has built for itself a position of immense power and responsibility – effectively a sovereign power, in fact, with which it can validate and rewrite the conditions for knowledge. It is important that the sciences of the artificial look at the nature of the new kinds of knowledge they generate and reflect them back to culture.

## References

- Agamben, G. (1998) *Homo Sacer: Sovereign Power and Bare Life* (trans. D. Heller-Roazen). Stanford University Press, Stanford.
- Agamben G. (1999) *The Man without Content* (trans. G. Albert). Stanford University Press, Stanford.
- Badiou A. (2001) *Ethics: An Essay on the Understanding of Evil* (trans. P. Hallward). Verso, London.
- Delanda, M. (2011). *Philosophy and Simulation: The Emergence of Synthetic Reason*. Continuum, London.
- Dokmaniç, I., Parhizkar, R., Walther, A., Lu, Y. M. and Vetterli, M (2013). Acoustic echoes reveal room shape. *Proc. National Academy of Sciences* June 17, 2013, doi:10.1073/pnas.1221464110.
- Impett, J. (2009). Making a mark: the psychology of composition. In Hallam, S., Cross, I. and Thaut, M., *The Oxford Handbook of Music Psychology*, pages 403-412. Oxford University Press, Oxford.
- Ingold, T. (2007). Against Soundscape. In Carlye, A., editor, *Autumn Leaves: Sound and the Environment in Artistic Practice*, pages 10-13. Double Entendre, Paris.
- Miranda, E. R., editor (2011). *A-Life for Music: Music and Computer Models of Living Systems*. A-R Editions, Middleton, WI.
- Miranda, E. R. and Biles, J., editors (2007). *Evolutionary Computer Music*. Springer, London.
- Moore, B (2002). Interference effects and phase sensitivity in hearing *Philosophical Transaction of the Royal Society. London A*, 360: 833-858.
- Ikeda, R. and Nicolai, C. (2011). *cyclo\_id*. Die Gestalten, Berlin.
- Viola, B. (2013). The Sound of One Line Scanning. In Lander, D. and Lexier, M., editors, *Sound by Artists*, pages 39-62. Charivari Press, Etobicoke, Ontario.

ECAL 2013 Submission 36, Programmable Nanomaterials Track.

# Programming the Assembly of Inorganic Nanomaterials Using Networked Chemical Reactions

Leroy Cronin, School of Chemistry, WestCHEM, The University of Glasgow, Glasgow, G12 8QQ, UK.

Email: [Lee.cronin@glasgow.ac.uk](mailto:Lee.cronin@glasgow.ac.uk); web: <http://www.croninlab.com>

**Keywords:** programmable nanomaterials; inorganic self assembly; inorganic reaction networks; chemical synthesis within 3d printed reactionware; programmable self assembly; emergent nanosystems

Traditionally much of nanochemical synthesis and self-assembly and transformations have been done in one pot reaction systems whereby reagents are added and the reaction conditions are controlled in terms of temperature, solvent, atmosphere and pressure. Although this general approach has been incredibly successful for the bottom up assembly of nanomaterials, the manipulation of programming of complex nanomaterials via reaction networks and the development of systems showing emergent properties require a fundamentally new approach, see Figures 1 and 2.

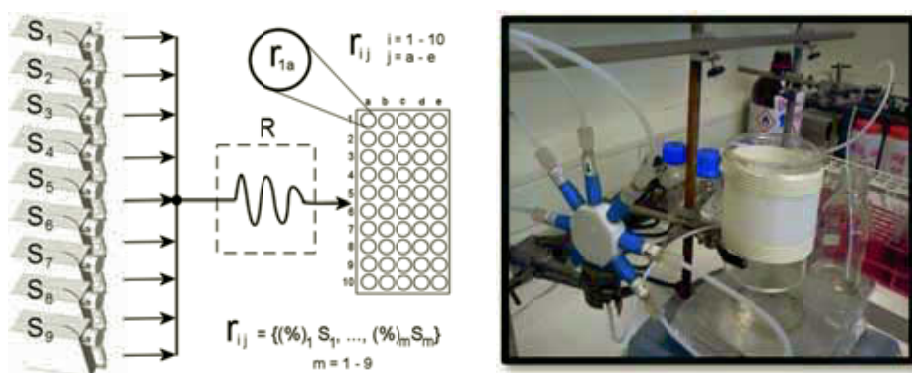


Figure 1. LEFT shows a schematic of the reaction array that can be set up by the continuous flow of reagent inputs ( $S_1$ - $S_9$ ) and the reaction array  $r_{ij}$  that results from the screening of the inputs against each other. RIGHT shows a photograph of the set up.

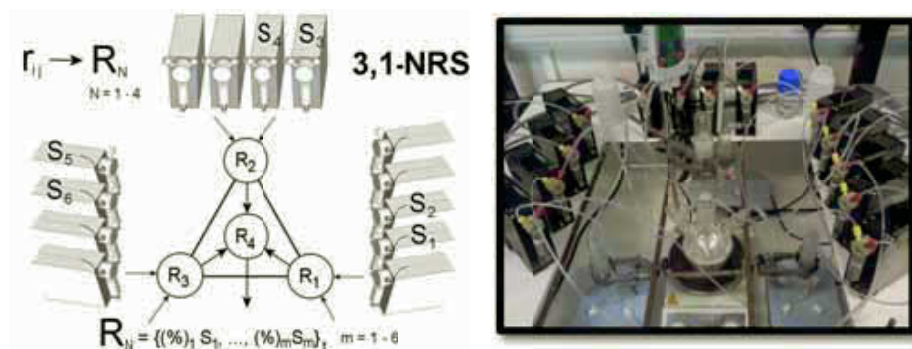


Figure 2. RIGHT shows a schematic of the networked reactor array (from reagent inputs  $S_n$ ) where different reactions  $R_n$  can be 'networked' or interconnected at different times during the process. RIGHT shows a photograph of the set up.

In this contribution I will outline our recent efforts, investigating the self-assembly and self-organization of inorganic nano molecules and the engineering of complex systems and reaction networks that lead to the emergence of system-level behaviours of interacting nanomolecules. To do achieve this we have developed new reaction techniques to control the assembly of nanoscale molecular metal oxide clusters, some of the largest non-biological molecules known, as well as new physical techniques e.g. the development of new cryospray and variable temperature mass spectrometry (for the elucidation of reaction mechanism and the observation of highly reactive intermediates). Ultimately our aim is to develop minimal inorganic systems capable of evolution, engineering materials with complex and emergent behaviours, as well as the development of new reaction formats for complex and novel chemistry e.g. flow systems and 3D-printing[1-5].

Acknowledgements: Andreu Ruiz de la Oliva, Victor Sans, Haralampos N. Miras

## References

1. H. N. Miras, G. J. T. Cooper, D.-L. Long, H. Bögge, A. Müller, C. Streb and L. Cronin, 'Unveiling the Transient Template in the Self Assembly of a Molecular Oxide Nano-Wheel', *Science*, 2010, 327, 72-74. DOI: 10.1126/science.1181735.
2. C. J. Richmond, H. N. Miras, A. R. de la Oliva, H. Zang, V. Sans, L. Paramonov, C. Makatsoris, R. Inglis, E. K. Brechin, D-L. Long, L. Cronin, 'A flow-system array for the discovery and scale up of inorganic clusters' *Nature Chem.*, 2012, 4, 1037-1043. DOI:10.1038/nchem.1489
3. I. Prat, J. S. Mathieson, M. Güell, X. Ribas, J. M. Luis, L. Cronin, M. Costas, 'Observation of Fe(V)=O using variable temperature mass spectrometry and its enzyme-like C–H and C=C oxidation reactions.' *Nature Chemistry*, 2011, 3, 788. DOI: 10.1038/nchem.1132
4. G. J. T. Cooper, P. J. Kitson, R. Winter, M. Zagnoni, D.-L. Long and L. Cronin 'Modular Redox Active Inorganic Chemical Cells: iCHELLs', *Angew. Chem. Int. Ed.*, 2011, 50, 10373. DOI: 10.1002/anie.201105068
5. M. D. Symes, P. J. Kitson, J. Yan, C. J. Richmond, G. J. T. Cooper, R. W. Bowman, T. Vilbrandt, L. Cronin, 'Integrated 3D printed reactionware for chemical synthesis and analysis', *Nature Chem.*, 2012, 4, 349-354. DOI :10.1038/nchem.1313

# Catalytic Molecular Walkers: Aspects of Product Release

Darko Stefanovic<sup>1</sup>, Milan N. Stojanovic<sup>2</sup>, Mark J. Olah<sup>1</sup> and Oleg Semenov<sup>1</sup>

<sup>1</sup>University of New Mexico, Albuquerque, NM 87131, USA

<sup>2</sup>Columbia University, New York, NY 10032, USA

darko@cs.unm.edu

## Abstract

We study a model of multi-legged catalytic molecular walkers that abstract a class of recently developed synthetic molecular motors. We focus on their kinetics of release of catalytic product, delineating the influence of chemical kinetic parameters, geometric configuration, loss from surface, and bulk kinetics. We show that such walkers can achieve a uniform rate of release over long time scales, which can be exploited in applications.

**ECAL Topics:** Molecular Motors; DNA Computing; Applications in Nanotechnology, Computable Matter, or Medicine.

## Introduction

Nanoscale objects are subject to random Brownian motion that leads to undirected diffusive transport. Nanoscale systems that require more precise control over the transport of materials and information must expend energy in some form to bias the random molecular motion in useful directions. A *molecular motor* is a nanoscale device capable of transforming chemical free energy into useful work and directed motion, and can function as a cargo transport device allowing superdiffusive transport of materials. Natural molecular motors play an important role in critical biological processes in the cell and are the source of most forms of motion in living beings (Schliwa and Woehlke, 2003; Vale and Milligan, 2000; Phillips et al., 2013).

In addition to naturally occurring molecular motors, several synthetic molecular motors have been designed recently (Yin et al., 2004; Bath and Turberfield, 2007; Muscat et al., 2011; Wickham et al., 2012; Shin and Pierce, 2004; Venkataraman et al., 2007; Omabegho et al., 2009; von Delius et al., 2010; Kay et al., 2007). We have synthesized catalytic DNA-based walkers called *molecular spiders* (Pei et al., 2006). A molecular spider has a rigid body and several flexible enzymatic legs that attach to and modify substrate DNA molecules (Figure 2a). When we first reported molecular spiders, we were keen to explore their motive properties and subsequently we experimentally demonstrated their ability to follow patterned nanoscale tracks of

DNA substrate molecules (Lund et al., 2010). Our mathematical modelling efforts similarly were focused on characterizing the spiders' walking gaits, such as predicting their ability to do organized mechanical work in opposition to an external force (which we hope experimentally to validate in the near future). On the other hand, in the original study we also observed excellent chemical processivity of the spiders; in one assay spiders and their substrates were randomly deposited in a matrix in a 1:3800 ratio, and the spiders cleaved nearly all substrates before dissociating from the matrix (Figure 1). We noted the nearly linear rate of substrate cleavage in this case, but this observation has not been explained or exploited.

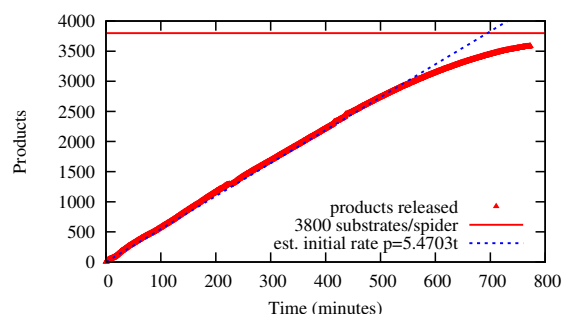


Figure 1: Data from a surface-plasmon resonance experiment. Four-legged spiders were released into a 2½D dextran matrix displaying the substrates. There were 3800 substrates to each spider, and over the course of the assay the spiders cleaved 100% of the substrate offered, exhibiting an initial linear increase in the amount of released product. (After Pei et al. (2006).)

In separate studies, we have shown that the deoxyribozyme chemistry of the spiders' legs is compatible with actuation, such as the release of small molecules (Kolpashchikov and Stojanovic, 2005; Yashin et al., 2007), as well as controllable by both oligonucleotides (Stojanovic et al., 2002) and small-molecule ligands (Stojanovic and Kolpashchikov, 2004), including via complex decision-



making molecular logic (Macdonald et al., 2006). Therefore, the sustained nearly uniform rate of release from a matrix, mediated by activated spiders (or other multivalent catalytic walker chemistries), may have promise in drug delivery. Indeed, achieving zero-order kinetics of drug release is a long-standing but difficult goal of pharmacokinetics (Siepmann et al., 1999; Siepmann and Peppas, 2000; Gupta and Kumar, 2000; Agrawal et al., 2006; Kim, 1995); a new study shows the use of nanoparticles to approximate it (Gu et al., 2013).

Here we numerically study the product-release properties of molecular spiders and show that:

1. walker-mediated release of product can be an effective tool for achieving a uniform rate of release over long time scales;
2. catalytic bias is essential for the operation of this mechanism;
3. catalytic bias is an effective tool for regulation of release rate;
4. multivalency is a practical mechanism for overcoming the problem of dissociation (confirming the results of Samii et al. (2010); Olah and Stefanovic (2013)).

The results provide a first guide to the vast parameter space available to the designers of potential molecular-spider-based drug delivery vehicles.

## Molecular Spiders and Their Models

In this paper we use a model that describes the motion of multipedal enzymatic walkers as they move over and modify chemical sites arranged as arbitrary 2D tacks and patterns (Figure 2b). A walker has a rigid body with attachment points for  $k \geq 2$  identical, flexibly tethered legs. Each leg has a reactive site at its foot that can reversibly bind to and enzymatically modify surface-bound chemical sites from a substrate species into a product species. As a walker moves over a surface of substrate sites, the enzymatic actions of the legs leave behind a trail of modified product sites. These modified sites create an asymmetrical distribution of substrates around the walker. When the leg-product binding is weaker than the leg-substrate binding, this local substrate asymmetry can bias the motion of the walker away from previously visited areas and towards unvisited sites, causing an otherwise unbiased symmetric walker to move superdiffusively. Thus, the walkers transform the chemical free energy in the surface sites into directional motion and can do ordered work in opposition to an external load force, hence they are a new type of molecular motor, useful for directional transport in nanoscale systems.

We formulate the model as a continuous-time Markov process that describes the stochastic motion of the walker as a sequence of transitions between discrete chemical states. Each walker state is described by the chemical state of the

surface sites (substrate or product) and the current set of sites where a leg is attached. The transitions from state to state correspond to the discrete chemical actions of legs binding to sites, unbinding from sites, or enzymatically modifying attached substrate sites to products. From any particular chemical state of the walker legs, each individual chemical action is independent of the previous state of the walker and of the chemical state of the other legs. In other words, the legs are not kinetically coupled. They are, however, mechanically constrained by their connection to a common body.

We use Monte Carlo algorithms to simulate the motion of multivalent random walkers as they move over a track of substrate molecules (Figure 2c). We find that when the rate of substrate catalysis is much slower than the rate of product detachment, the walkers (on average) move superdiffusively away from the origin. Furthermore, our model is designed to quantify the effect of a constant load force opposing the walker motion. We have found (Olah and Stefanovic, 2013) that the superdiffusive motion of the walker persists even under moderate loads of  $\leq 2.0$  pN, for legs with maximum extension length  $\ell = 12.5$  nm ( $2.5 \times$  the lattice pitch). Hence, the walkers are capable of transforming the chemical free energy of substrate molecules into directed transport and ordered mechanical work. We designed the model to investigate a minimal set of mechanical and kinetic features that are necessary to transform the otherwise unbiased diffusive motion of multipedal enzymatic walkers into directional, superdiffusive motion useful for nanoscale cargo transport: even though a walker's legs are identical, unoriented, uncoordinated, and unconstrained other than the passive constraint implied by their connection to a common body, nonetheless we find that these simple geometric constraints on the legs combined with the kinetic bias in the direction of unvisited substrates suffice to generate superdiffusive motion, even when that motion is opposed by an external load force.

While this model provides good physical detail, it can be computationally rather demanding, which justifies mathematical models of molecular spiders at greater levels of abstraction, for instance, simplifying the chemical kinetics and the geometry of the walker and its track. Antal and collaborators studied a single simplified spider on a one-dimensional track (Antal et al., 2007; Antal and Krapivsky, 2007), showing that a difference in residence time between substrates and products, in conjunction with the presence of multiple legs, biases the motion towards fresh substrates when the spider is on a boundary between substrates and products. We showed that this bias makes spiders move superdiffusively for long periods of time (Semenov et al. (2011a). Samii et al. investigated various gaits and numbers of legs (Samii et al., 2010, 2011), emphasizing the possibility of detachment from a 1D track. We studied the behavior of multiple spiders continuously released onto a 1D track (Semenov et al., 2011b, 2012) from a point source. Re-

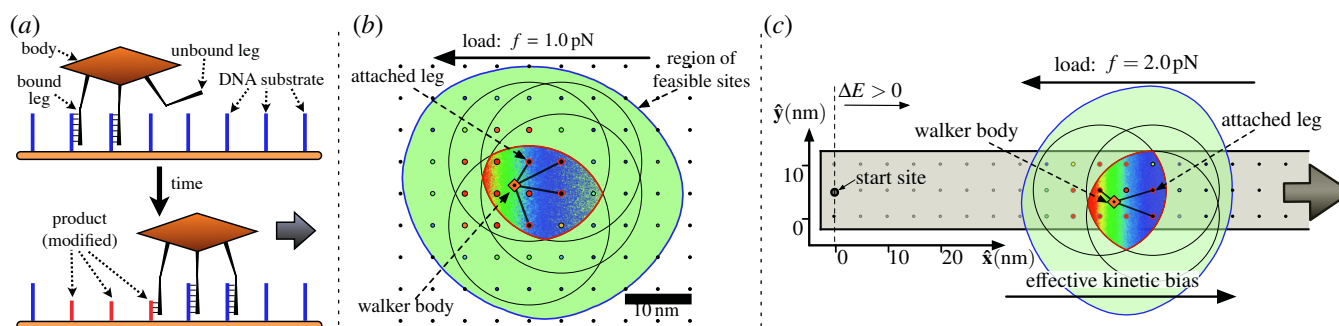


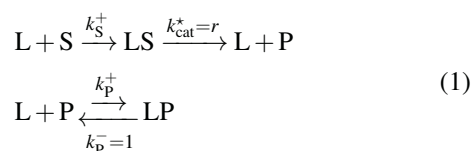
Figure 2: The multivalent random walker model describes the motion of multipedal walkers acting as molecular motors as they move over tracks of surface-bound substrate fuel. (a) The model is inspired by a type of DNA nanowalker called a *molecular spider* that has a rigid, inert body and several enzymatic legs that can attach to and cleave complementary DNA substrates arrayed as nanoscale tracks. (b) The model describes the 2D motion of molecular-spider-like walkers. We assume the mechanical motion of the body and of its legs comes to an equilibrium in between chemical steps, allowing us to describe the attachment likelihood over the set of local feasible attachment sites. (c) Under appropriate kinetic biases between product and substrate sites, the walkers move superdiffusively in the transient, even when their motion is opposed by a conservative load force.

lated to the present topic, Antal and Krapivsky evaluated the diffusion constant and the amplitude describing the asymptotic behavior of the number of visited sites (i.e., released products) for a single simplified spider on an infinite square lattice (Antal and Krapivsky, 2012).

### Model Definition

The formal model we assume here is a minor simplification of our general model, described in detail in Olah and Stefanovic (2013). In the context of a product-release application, as opposed to transport, we need not be concerned with external forces, and therefore we can set them to zero, which makes all feasible body positions equally likely; this eliminates the need for expensive equilibrium Monte Carlo computations in the inner loop of the simulation. Further, we approximate the region of feasible sites (Figure 2(b)) as the intersection of circles of radius  $2\ell$  around all current attachment points; this eliminates the need for elaborate computational geometry calculations. Finally, we assume all sites within the feasible region are equally likely to be chosen as the next attachment point for the leg that is moving.

The chemical reaction scheme is



showing the essentially irreversible binding of an enzymatic leg  $L$  to a substrate  $S$  to form a complex  $LS$ , which leads irreversibly to the cleavage of the surface-bound substrate into two product parts. The scheme only shows the part  $P$  that remains bound to the surface; the other is released into

the environment, and is in fact the product part of interest in applications. The second reaction describes the fleeting, reversible, binding of a leg to the surface-bound products. The rate  $r$  encapsulates both the catalytic cleavage of the substrate and the subsequent dissociation of the leg from the products. The rate of dissociation from products is set to 1, i.e., our model is dimensionless.

We use kinetic Monte Carlo simulation (Bortz et al., 1975; Voter, 2007) to generate trajectories of the model for various parameter combinations. The implementation techniques we use are standard; of note, the results in this paper were generated with a highly customizable representation written in Haskell.

### Results

We have obtained simulation results for the following product release scenario. The walking surface contains 40,000 sites in a  $200 \times 200$  box in a 2D square lattice, initially consisting entirely of substrates. One hundred walkers are deposited uniformly at random onto the surface, each walker having one leg attached to a substrate and the remaining legs free. This gives a 1:400 ratio of walkers to substrates, a scaled-down version of the laboratory experiments. The walkers are followed as they move around the box until all substrates have been turned into products.

We observe the number of products on the surface, which is equal to the number of products released into the surrounding medium. We do not model the diffusion of released products into the environment. To control for the possibility of walker dissociation, we also observe the number of walkers left on the surface.

We study the effects of spider geometry by varying the

number of legs from 1 to 6. In the plots that follow, only 2-, 4-, and 6-legged walkers are shown, for clarity. One-legged walkers dissociate from the surface immediately; 3- and 5-legged walkers have intermediate behaviors as expected. We also vary the leg length. We use the length  $\ell$  equal to 2.5 times the lattice pitch as our baseline, and also consider what happens if the legs are much shorter,  $\ell = 1$ , or longer,  $\ell = 5$ .

We study the effects of different chemical kinetics by varying the catalytic rate parameter  $r$ , i.e., the degree of residence-time bias between substrates and products. We also vary the binding rate parameters  $k_S^+$  and  $k_P^+$ .

### Baseline observations

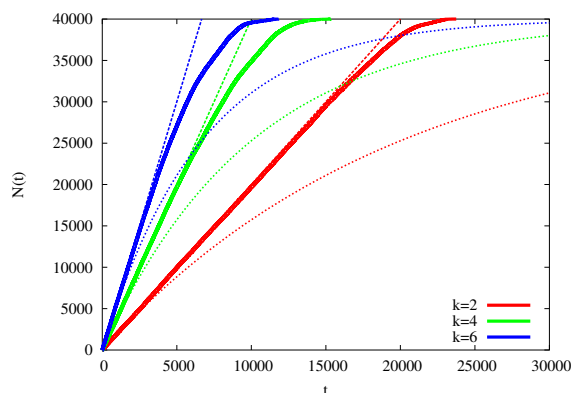


Figure 3: Number of products released as a function of time for 2-, 4-, and 6-legged walkers, with  $r = 0.01$ ,  $k_P^+ = 1000.0$ ,  $k_S^+ = 1000.0$ , and  $\ell = 2.5$ .

Shown in Figure 3 are plots for product release using a baseline set of parameter values (Olah and Stefanovic, 2013),  $r = 0.01$ ,  $k_P^+ = 1000.0$ ,  $k_S^+ = 1000.0$ ,  $\ell = 2.5$ , and for three different leg configurations, 2-, 4-, and 6-legged.

Consider the product release curve for the four-legged walkers (green). It rises essentially linearly for a long time before tapering off towards the ceiling of 40,000 as eventually all sites are turned into products. In this respect the model captures the empirical observations from Figure 1. Also drawn in green are two theoretical curves. The dashed line corresponds to exactly linear product release at a rate equal to the initial rate for the walker (zero-order kinetics). Note that the initial rate of product release for the walker is easily seen to be the product of the substrate catalysis-dissociation rate, which we are calling  $r$ , and the number of legs, since almost immediately upon deposition all legs are on substrates. The dotted line corresponds to an exponential decay that starts with an initial rate equal to the walker's, and tends to the same asymptotic value (first-order kinetics). The observed curve is bracketted between the two theoretical curves, adhering more closely to zero-order kinetics.

Analogous families of curves are shown for two-legged

walkers (red) and six-legged walkers (blue). The initial slope is proportional to the number of legs because each leg is an independent enzyme, and initially all legs quickly find a substrate to bind to. Interestingly, the adherence of the observed kinetics to zero-order kinetics lasts longer for  $k = 2$  walkers than for walkers with more legs.

### Varying the catalysis kinetics

Here we show the effects of varying the parameter  $r$ , keeping the binding kinetics fixed at the value  $k_P^+ = 1000.0$ ,  $k_S^+ = 1000.0$ , which means that binding is very fast and there is no binding preference between substrates and products. Figures 4(a)–(c) display  $r = 0.001, 0.1, 1.0$ , in addition to  $r = 0.01$  from Figure 3.

The initial rate of product release varies in proportion to  $r$ , as expected because  $r$  is the hopping rate for legs on substrates. The overall time to completion grows roughly in inverse proportion to  $r$ , therefore the four plots are accordingly scaled, allowing us to observe the changes in the shapes of the curves. The adherence of the observed product release curve to the zero-order kinetics theoretical curve improves when  $r$  is decreased to 0.001 and it worsens when  $r$  is increased. When  $r$  reaches unity, and the residence-time bias is lost, product release becomes slower and less uniform even than first-order kinetics; furthermore, there is an inversion effect for the number of legs—even though initial product release rates are higher for greater  $k$ , eventually these walkers fall behind two-legged ones.

### Varying the binding kinetics

In these simulations, we keep the catalysis fixed at the baseline value of  $r = 0.01$  but we vary the binding kinetics, i.e., the on-rates  $k_P^+$  and  $k_S^+$ .

In Figure 5, we decrease the on-rate for substrates tenfold, so that a leg is effectively repelled from substrates. Comparing with the baseline results of Figure 3, while the initial rates (dominated by catalytic cleavage) are not affected, the product release curves diverge from zero-order kinetics much sooner, presumably because the walkers are diverted from exploration of fresh substrates and find themselves diffusing in their own local seas of product. If we also decrease the on-rate for products we recover the original behavior as in Figure 3 (we reserve for future study the effects of decreasing the two on-rates further towards the catalysis rate, outside of the regime of rates measured for deoxyribozyme walkers).

### Varying the leg length

Keeping the chemical kinetics parameters at the baseline, we now vary the geometry of the walker, namely its leg length. The baseline leg length  $\ell = 2.5$  corresponds to the molecular spiders used in laboratory experiments, Figure 3. If the legs are made much shorter, as in Figure 6(a), spiders with

too many legs become slower. A detached leg is too constrained by the maximum distance from the remaining legs' attachment points, and it is impeded from finding fresh substrates. Thus the product release of the 4-legged walker is slower than first-order kinetics, but for the 6-legged walker it is slowed down to even less than the 2-legged walkers' pace.

On the other hand, if the legs are made longer, as in Figure 6(b), a moving leg has a greater probability of finding a substrate when the walker is on the edge of the patch it has grazed, hence we observe better adherence to zero-order product release kinetics.

## Dissociation

Since in our walker model the legs are uncoordinated, it is possible for all legs of a walker to become detached. When that happens, the walker is removed from the simulation—indeed in control experiments Pei et al. (2006) we had measured the rates at which dissociation happened with molecular spiders. However, in the model as well as in the laboratory, dissociation is a rare event for the chosen baseline kinetics. In Figure 7 we show the number of walkers present, initially 100. Dissociation is a significant phenomenon only for two-legged walkers; for walkers with more legs, simultaneous detachment is unlikely. Thus, dissociation is not the primary cause of the drop-off in the product release rate; rather increased dissociation is a consequence of the decay into diffusion over products as each walker's local area of products grows, or as the neighboring areas merge: legs on products detach much more frequently and thus the rate at which dissociation events occur is much greater.

## Discussion

Let us consider the model of Equation 1 as if occurring in bulk solution, where the initial number of substrate molecules is the same as the initial number of sites (40,000), and the initial number of enzymes is equal to the total number of spiders' legs (200 for two-legged spiders). As shown in Figure 8(a), for  $r = 0.01$  bulk release (black curve) is nearly linear. On the other hand, in Figure 8(b) for  $r = 1$  bulk release amounts to exponential decay. Thus, as  $r$  is decreased from 1 towards 0, bulk release improves from first-order kinetics to zero-order kinetics. Spider release is also shown, and it is in both cases slower than the corresponding bulk release. After all, this should be expected from first principles. (1) In a well-mixed solution, all potential substrates are available to the enzyme. With spiders, this is not the case because substrates are a local resource; whenever the spider is diffusing in the sea of products, they are not available at all. (2) Eventually some spiders completely dissociate. These two effects make spider product release less efficient than with the hypothetical bulk solution reaction. Both these effects diminish as  $r$  is decreased from 1 towards 0. This explains why, for small values of  $r$ , product release

by spiders is nearly the same as bulk release, which in turn is nearly linear. Thus, under suitable conditions, spiders are an effective mechanism that allows an enzyme to visit surface-immobilized substrates, yielding almost the same resultant kinetics as in a perfectly mixed volume reaction.

Here we have studied a practically motivated problem: the release of products in a bounded domain by a fixed finite number of walkers. In future work, we will explore the behavior of walkers in an unbounded domain, such as the plane. In this setting, there are known analytical results for ordinary random walkers and for the simplest spider models Antal and Krapivsky (2012), namely that asymptotically  $N(t) \propto \frac{t}{\log t}$ , and we shall examine how the catalysis and the presence of multiple legs affect this behavior. It will be interesting to connect the unbounded-domain behaviors with a more detailed analysis of the bounded-domain cases, including varying how the walkers are initially deposited (different spatial distributions), and considering sources of injection.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under grants 1027877 and 1028238.

## References

- Agrawal, S. K., Sanabria-DeLong, N., Coburn, J. M., Tew, G. N., and Bhatia, S. R. (2006). Novel drug release profiles from micellar solutions of plapeopla triblock copolymers. *Journal of Controlled Release*, 112:64–71.
- Antal, T. and Krapivsky, P. L. (2007). Molecular spiders with memory. *Physical Review E*, 76(2):021121.
- Antal, T. and Krapivsky, P. L. (2012). Molecular spiders on a plane. *Physical Review E*, 85:061927.
- Antal, T., Krapivsky, P. L., and Mallick, K. (2007). Molecular spiders in one dimension. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(08):P08027.
- Bath, J. and Turberfield, A. J. (2007). DNA nanomachines. *Nature Nanotechnology*, 2(5):275–284.
- Bortz, A. B., Kalos, M. H., and Lebowitz, J. L. (1975). A new algorithm for Monte Carlo simulation of Ising spin systems. *Journal of Computational Physics*, 17(1):10–18.
- Gu, Z., Aimetti, A. A., Wang, Q., Dang, T. T., Zhang, Y., Veisoh, O., Cheng, H., Langer, R. S., and Anderson, D. G. (2013). Injectable nano-network for glucose-mediated insulin delivery. *ACS Nano*. Published online 10.1021/nn400630x.
- Gupta, K. C. and Kumar, M. N. V. R. (2000). Drug release behavior of beads and microgranules of chitosan. *Biomaterials*, 21:1115–1119.
- Kay, E. R., Leigh, D. A., and Zerbetto, F. (2007). Synthetic molecular motors and mechanical machines. *Angewandte Chemie International Edition*, 46(1-2):72–191.



- Kim, C. (1995). Compressed donut-shaped tablets with zero-order release kinetics. *Pharmaceutical Research*, 12(7):1045–1048.
- Kolpashchikov, D. M. and Stojanovic, M. N. (2005). Boolean control of aptamer binding states. *Journal of the American Chemical Society*, 127(32):11348–11351.
- Lund, K., Manzo, A. J., Dabby, N., Michelotti, N., Johnson-Buck, A., Nangreave, J., Taylor, S., Pei, R., Stojanovic, M. N., Walter, N. G., Winfree, E., and Yan, H. (2010). Molecular robots guided by prescriptive landscapes. *Nature*, 465:206–210.
- Macdonald, J., Li, Y., Sutovic, M., Lederman, H., Pendri, K., Lu, W., Andrews, B. L., Stefanovic, D., and Stojanovic, M. N. (2006). Medium scale integration of molecular logic gates in an automaton. *Nano Letters*, 6(11):2598–2603.
- Muscat, R. A., Bath, J., and Turberfield, A. J. (2011). A programmable molecular robot. *Nano Letters*, 11(3):982–987.
- Olah, M. J. and Stefanovic, D. (2013). Superdiffusive transport by multivalent molecular walkers moving under load. *arXiv:1211.3482 [physics.bio-ph]*.
- Omabegho, T., Sha, R., and Seeman, N. C. (2009). A bipedal DNA brownian motor with coordinated legs. *Science*, 324(5923):67–71.
- Pei, R., Taylor, S. K., Stefanovic, D., Rudchenko, S., Mitchell, T. E., and Stojanovic, M. N. (2006). Behavior of polycatalytic assemblies in a substrate-displaying matrix. *Journal of the American Chemical Society*, 128(12):12693–12699.
- Phillips, R., Kondev, J., Theriot, J., and Garcia, H. G. (2013). *Physical Biology of the Cell*. Garland Science, London and New York, 2nd edition.
- Samii, L., Blab, G. A., Bromley, E. H. C., Linke, H., Curmi, P. M. G., Zuckermann, M. J., and Forde, N. R. (2011). Time-dependent motor properties of multipedal molecular spiders. *Phys. Rev. E*, 84:031111.
- Samii, L., Linke, H., Zuckermann, M. J., and Forde, N. R. (2010). Biased motion and molecular motor properties of bipedal spiders. *Phys. Rev. E*, 81:021106.
- Schliwa, M. and Woehlke, G. (2003). Molecular motors. *Nature*, 422(6933):759–765.
- Semenov, O., Olah, M., and Stefanovic, D. (2012). Cooperative linear cargo transport with molecular spiders. *Natural Computing*, pages 1–18.
- Semenov, O., Olah, M. J., and Stefanovic, D. (2011a). Mechanism of diffusive transport in molecular spider models. *Physical Review E*, 83(2):021117.
- Semenov, O., Olah, M. J., and Stefanovic, D. (2011b). Multiple molecular spiders with a single localized source—the one-dimensional case. In *DNA 17: Proceedings of The Seventeenth International Meeting on DNA Computing and Molecular Programming*, volume 6397 of *Lecture Notes in Computer Science*, pages 204–216. Springer.
- Shin, J.-S. and Pierce, N. A. (2004). A synthetic DNA walker for molecular transport. *Journal of the American Chemical Society*, 126(35):10834–10835.
- Siepmann, J., Kranz, H., Bodmeier, R., and Peppas, N. A. (1999). HPMC-matrices for controlled drug delivery: A new model combining diffusion, swelling, and dissolution mechanisms and predicting the release kinetics. *Pharmaceutical Research*, 16(11):1748–1756.
- Siepmann, J. and Peppas, N. A. (2000). Hydrophilic matrices for controlled drug delivery: An improved mathematical model to predict the resulting drug release kinetics (the “sequential layer” model). *Pharmaceutical Research*, 17(10):1290–1298.
- Stojanovic, M. N. and Kolpashchikov, D. M. (2004). Modular aptameric sensors. *Journal of the American Chemical Society*, 126(30):9266–9270.
- Stojanovic, M. N., Mitchell, T. E., and Stefanovic, D. (2002). Deoxyribozyme-based logic gates. *Journal of the American Chemical Society*, 124(14):3555–3561.
- Vale, R. D. and Milligan, R. A. (2000). The way things move: Looking under the hood of molecular motor proteins. *Science*, 288(5463):88–95.
- Venkataraman, S., Dirks, R. M., Rothmund, P. W. K., Winfree, E., and Pierce, N. A. (2007). An autonomous polymerization motor powered by DNA hybridization. *Nature Nanotechnology*, 2:490–494.
- von Delius, M., Geertsema, E. M., and Leigh, D. A. (2010). A synthetic small molecule that can walk down a track. *Nature Chemistry*, 2(2):96–101.
- Voter, A. (2007). Introduction to the kinetic Monte Carlo method. In Sickafus, K., Kotomin, E., and Uberuaga, B., editors, *Radiation Effects in Solids*. Springer.
- Wickham, S. F. J., Bath, J., Katsuda, Y., Endo, M., Hidaka, K., Sugiyama, H., and Turberfield, A. J. (2012). A DNA-based molecular motor that can navigate a network of tracks. *Nature Nanotechnology*, 7(3):169–173.
- Yashin, R., Rudchenko, S., and Stojanovic, M. N. (2007). Networking particles over distance using oligonucleotide-based devices. *Journal of the American Chemical Society*, 129(50):15581–15584.
- Yin, P., Yan, H., Daniell, X. G., Turberfield, A. J., and Reif, J. H. (2004). A unidirectional DNA walker that moves autonomously along a track. *Angewandte Chemie International Edition*, 43:4906–4911.

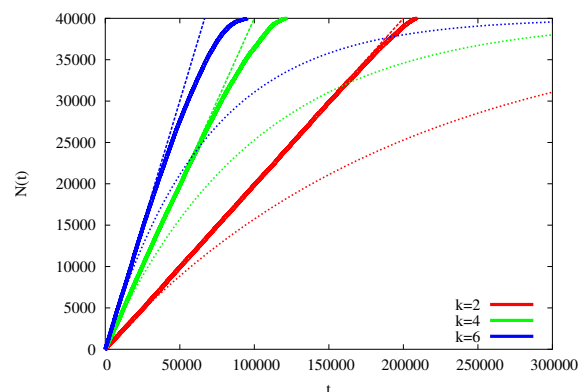
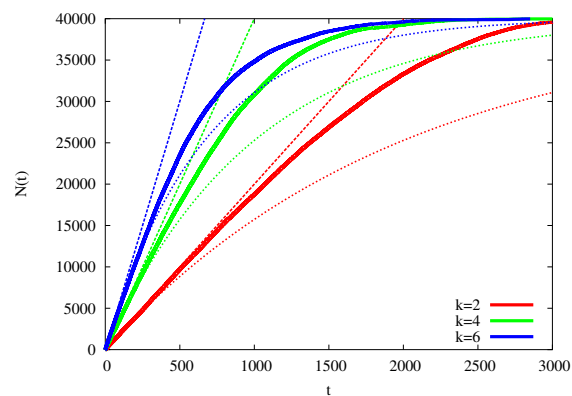
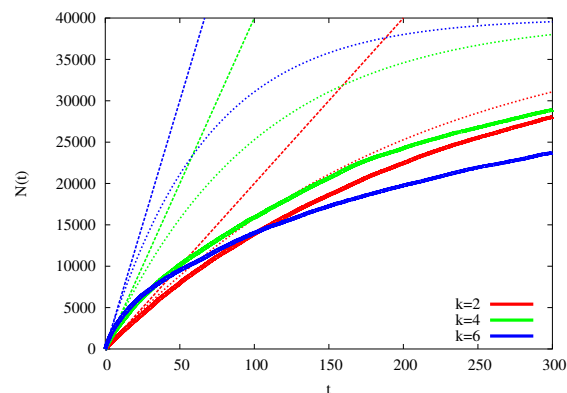

 (a)  $r = 0.001$ 

 (b)  $r = 0.1$ 

 (c)  $r = 1.0$ 

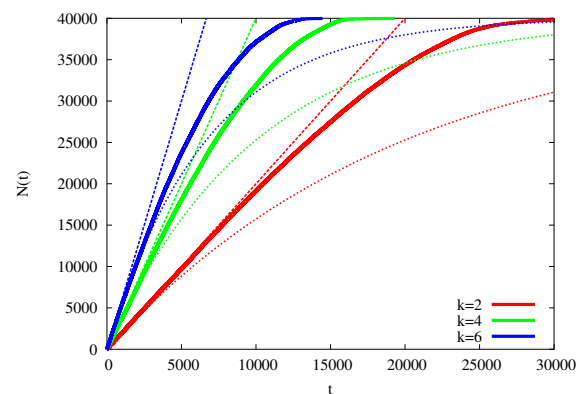
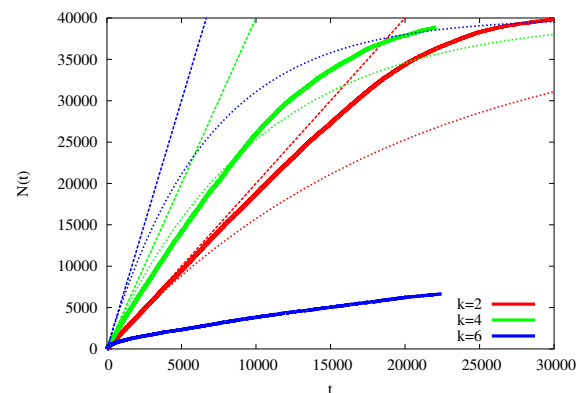
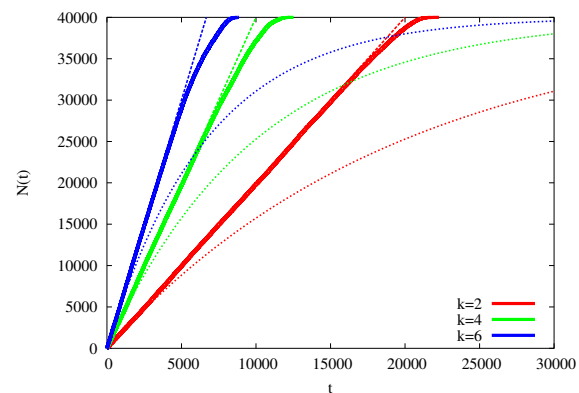
 Figure 4: Varying the catalytic rate  $r$ : number of products released as a function of time for 2-, 4-, and 6-legged walkers.

 Figure 5: Varying the binding kinetics—“repellent” substrates,  $k_p^+ = 1000$ ,  $k_s^+ = 100$ : number of products released as a function of time for 2-, 4-, and 6-legged walkers.

 (a) Short legs,  $\ell = 1.0$ 

 (b) Long legs,  $\ell = 5.0$ 

Figure 6: Varying the leg length: number of products released as a function of time for 2-, 4-, and 6-legged walkers.

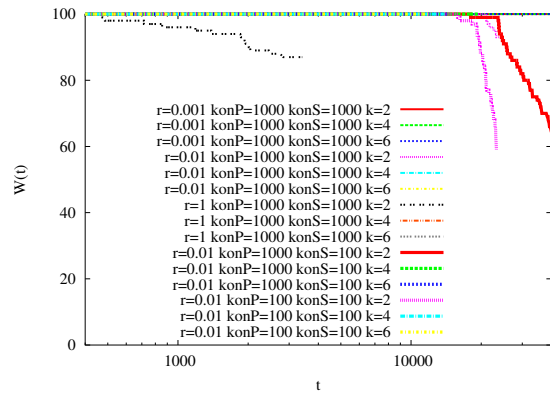
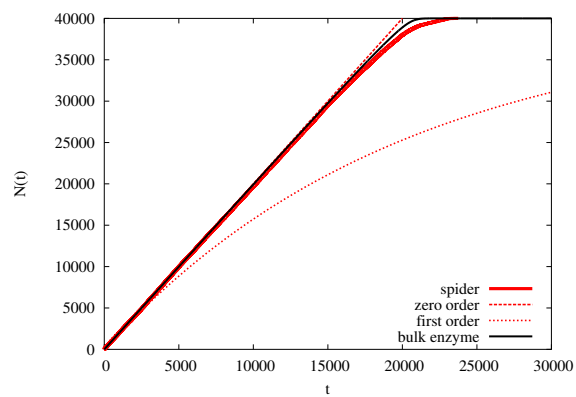
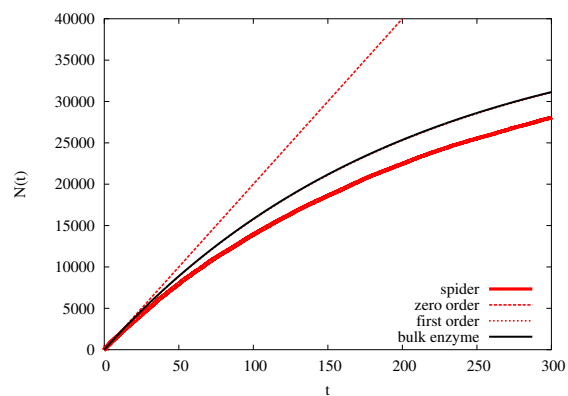


Figure 7: Number of walkers remaining as a function of time for selected configurations.



(a)  $r = 0.01$



(b)  $r = 1.0$

Figure 8: Comparison with bulk reaction: number of products released as a function of time, for 2-legged walkers.

ECAL 2013 Submission 38, Synthetic and Systems Biochemistry and Biological Control

Synthetic biology manipulations in 3D printed wet-ware

Leroy Cronin, School of Chemistry, WestCHEM, The University of Glasgow, Glasgow, G12 8QQ, UK.

Email: [Lee.cronin@glasgow.ac.uk](mailto:Lee.cronin@glasgow.ac.uk); web: <http://www.croninlab.com>

Keywords: synthetic biology; flow systems; fluidics; 3d printing; protein expression; cell free; gene transfer

In our laboratory we have been developing new approaches to discover the 'transition-to-evolvability' in chemistry. This is because if we can discover or engineer an abiotic system that can evolve (we could define this as an inorganic chemical cell -iCHELL) we might be able to suggest that synthetic biology can exist in many chemical forms, of which the terrestrial biology found on planet earth is one subset. It could even help us establish the idea that evolvability is the key signature that defines living from non-living systems. This problem is rather vast since our aim is to compress a planet sized reaction vessel and a 400 M year run-time into a laboratory over a few years! Not only does this extraordinary problem require new radical chemical approaches[1], it also requires the development of some radical new technological solutions[2-3]. In this talk I will cover both aspects with an emphasis on how some of our new approaches can be applied to both 'inorganic' and 'organic' synthetic biology with a special emphasis on rapid fabrication of systems for synthetic biology that allows the 'plug and play' of synthetic biology in new fluidic formats, see Figure [1-2].

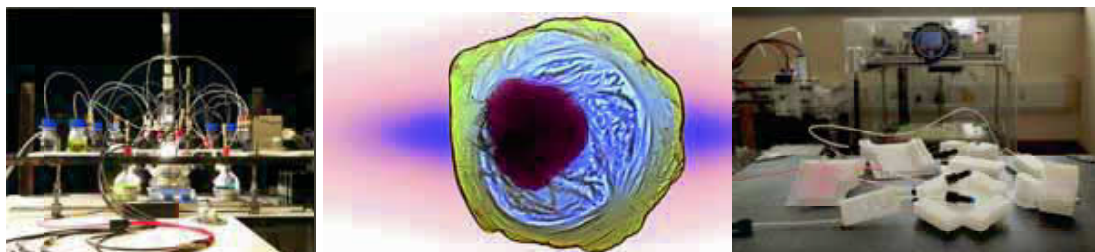


Figure. Images depicting our different approaches and technologies required for our quest: Networks, iCHELLs and 3D printing of chemical reactionware and biochemical bioware.

1. C. J. Richmond, H. N. Miras, A. R. de la Oliva, H. Zang, V. Sans, L. Paramonov, C. Makatsoris, R. Inglis, E. K. Brechin, D-L. Long, L. Cronin, 'A flow-system array for the discovery and scale up of inorganic clusters' *Nature Chem.*, 2012, 4, 1037-1043. DOI:10.1038/nchem.1489
2. M. D. Symes, P. J. Kitson, J. Yan, C. J. Richmond, G. J. T. Cooper, R. W. Bowman, T. Vilbrandt, L. Cronin, 'Integrated 3Dprinted reactionware for chemical synthesis and analysis', *Nature Chem.*, 2012, 4, 349-354. DOI :10.1038/nchem.1313



## Evolving cheating DNA networks: a Case Study with the Rock–Paper–Scissors Game

Nathanaël Aubert<sup>1</sup>, Quang Huy Dinh<sup>1</sup>, Masami Hagiya<sup>1</sup>, Teruo Fujii<sup>3</sup>,  
Hitoshi Iba<sup>2</sup>, Nicolas Bredeche<sup>4,5</sup>, Yannick Rondelez<sup>3</sup>

<sup>1</sup> Graduate School of Information Science and Technology, The University of Tokyo, Japan

<sup>2</sup> Graduate School of Engineering, The University of Tokyo, Japan

<sup>3</sup> LIMMS/CNRS-IIS, Institute of Industrial Science, The University of Tokyo, Japan

<sup>4</sup> UPMC Univ Paris 06, UMR 7222, ISIR, F-75005, Paris, France <sup>5</sup> CNRS, UMR 7222, ISIR, F-75005, Paris, France

naubert@is.s.u-tokyo.ac.jp

### Abstract

In models of games, the indirect interactions between players, such as body language or knowledge about the other's playstyle, are often omitted. They are, however, a rich source of information in real life, and increase the complexity of possible strategies. In the game of rock-paper-scissors, the simple monitoring of the opponent's move *before* it was played is a sufficient condition to trigger an arms race of detection and misinformation among evolved individuals. The most interesting aspect of those results is that they were obtained by evolving purely chemical reaction networks thanks to an adapted version of the famous NEAT algorithm. More specifically, those individuals were represented as biochemical systems built on the DNA toolbox, a paradigm that allows both easy *in-vitro* implementation and predictive *in-silico* simulation. This guarantees that the specific motives that emerged in this competition would behave identically in a test tube, and thus can be used in a more generic context than the current game.

### Introduction

The game of rock-paper-scissors, while being simple, can actually lead to interesting dynamics when it is played multiple times in a row. In particular, each player will try to “read” their opponents in the hope of getting the upper hand. However, if psychological factors are not taken into account, that is, if players are purely logical, game theory predicts that after a while, the optimal strategy becomes to play randomly with no bias among the three possible moves (Smith, 1993). Variations of the basic rules exist, but are expected to display the same kind of behaviors (from the point of view of game theory) as the classic three moves.

Interestingly, this game can be a good description of many mechanisms ranging from reproductive strategies of some species of lizards (Sinervo and Lively, 1996) or bacteria (Kerr et al., 2002) to oscillations in a gene regulatory circuit (Elowitz and Leibler, 2000). In all cases, there are three possible moves, each strong against another and weak against the remaining one. This usually leads to dynamical behaviors where the different players are constantly invading each other, forming complex spiral structures in two dimensional systems (Kerr et al., 2002; Reichenbach et al., 2007). Even

real life examples, such as the lizard example, display oscillations in population size, with a turnover of approximately six years, based on the field data of (Sinervo and Lively, 1996). Those dynamics may degenerate into a uniform population depending on the initial conditions, or such parameters as the mobility of the players. On the other hand, they may also occur even in a well-mixed system, where there is no spatial compartmentalization to protect diversity, if a given move gets stronger when it is less frequent (Freen and Abraham, 2001) or if the system never stalls, like in the repressilator (Elowitz and Leibler, 2000).

However, all those examples either suppose or require that a given individual will always “play” the same move. Indeed, the lizard will always have the same size and coloration, bacteria the same genotype and genes in the repressilator are not expected to arbitrarily change which target genes they inhibit. From a strategic point of view, more possibilities open when each agent can decide, at each time, which move he wants to put forward. In such a case, some form of knowledge of the opponent becomes necessary in order to infer his probable next move and play accordingly. This knowledge is obtained from two sources: cheating and analysis of the opponent previous moves. “Cheating” here designates the fact of obtaining clues about an opponent from its behavior just prior to the game, not in the negative sense of making a game uninteresting by bypassing the rules. Note that cheating in this sense is both an integral part of most human plays and of biological strategies, and in any way is an essential ingredient of any physically instantiated game. In fact, instantaneous moves and decisions are not possible in a physical world, which means that information is always leaked somehow. This fact was used by the Ishikawa laboratory in Japan to program a robot hand (Namiki et al., 2003) reacting fast enough to hand gestures to be able to always win against a human (video online).

While both cheating and strategic analysis requires significant abilities and are generally associated with intelligent players (or at least, players with intents), we wanted to demonstrate in this paper that purely molecular systems are also capable of intricate strategies, whose complexity can

be comparable to that of real players. Indeed it has been recently demonstrated that Turing universality can be achieved through the sole use of chemical reactions (Magnasco, 1997; Soloveichik et al., 2010; Cardelli, 2011). Moreover, practical bottom-up approaches have been proposed to actually instantiate arbitrary reaction networks (Seelig et al., 2006; Qian and Winfree, 2011). However, experimentally, only relatively simple tasks (equivalent in complexity to those performed by the most basic electronic circuits) have been demonstrated. Even from a theoretical standpoint only quite simple systems have been proposed, very far from the intricacy observed in the case of cellular regulation maps, or even bacterial behaviors.

The individuals we evolved were defined as entities from the DNA toolbox (Montagne et al., 2011), a particular paradigm to define DNA-based computing systems. In particular, we build on a unique feature of the DNA toolbox, which is to couple a generalized experimental strategy for the *in vitro* building of reaction networks to the availability of straightforward (if large, from the point of view of equation solving) quantitative models. These models allows exact mathematical predictions and thus allow to perform both *in vitro* and *in silico* designs in parallel.

Individuals were evolved through an adapted version of NeuroEvolution of Augmenting Topologies (NEAT) (Stanley and Miikkulainen, 2002), dubbed bioNEAT, using a fitness function based on how well they fared in a population-wide tournament. To our surprise, the apparition of a basic memory was not hard, but was almost immediately discarded, as it was not able to compete against cheating. Due to the necessity of having both players in the same well-mixed environment, it was much more efficient for an individual to actually develop a way to monitor the actions of its opponent while hiding its own move. When pushed to the extreme, this strategy produced interesting dynamics where individuals went through multiple moves before the end of the countdown, trying to settle into a winning position, eventually leading to some fashion of oscillatory systems. The mechanisms used for those purpose were interesting in themselves, including concentration comparators or system with multiple levels of activation, giving, through motif mining, insight into the possibilities of the DNA-toolbox. This showed that indeed, the behavior of purely molecular systems, corresponding to a realistic, directly implementable chemistry, can be interpreted in terms of complex strategic planning.

## Related Work and Current Contributions

Our work builds on multiple sources since it mixes design by genetic algorithm with molecular programming. Game theory was also an important source of inspiration, and was useful to check that our evolved individuals are playing in a way that differs from hypothetical “perfect” players.

## Rock-paper-scissors

There are also many previous works related to the game of rock-paper-scissors. However, to the best of our knowledge, they either use individuals which are only capable of playing one move, or link existing dynamics to an instance of the game. The evolution game theory study in (Smith, 1993) is the closest to our work, but lacks the added dimension that comes with dealing with cheating or leak of information (Cook et al., 2012). While DNA-based systems can hardly be described as having any form of intelligence, it is easy to rationalize their behavior as cheating, a very real possibilities among human players that is not taken into account in (Smith, 1993).

## Motif Mining

The idea of using DNA computing to play games has been previously introduced (Macdonald et al., 2008). Finding systems able to play a game is in itself a challenge that leads to developing new structures, and potentially solve issues related to real life problems. However, the use of evolutionary algorithms (Eiben and Smith, 2003) stand as a promising candidate to search for interesting reaction circuits. From the structural point of view, the analysis of the fittest individuals of specific runs revealed common functional motifs, which may help build new systems. This is the fundamental approach of synthetic biology, in which biological modules are recombined to perform engineered operations (Purnick and Weiss, 2009). In particular, it was interesting to note that, although actual patterns may vary from individuals to individuals, it was possible to classify them into rough generic categories. This could be used to create minimal libraries of structures for dynamic systems, that is, off-the-shelves building blocks like those defined in (Rodrigo et al., 2011). Such libraries would in turn allow the fast and reliable development of complex DNA-based systems. While, in our case, the structures evolved by the algorithm are possibly not generic enough to be useful in any given context, they still have potential applications for the design of a variety of such systems.

## Model

### The DNA toolbox

The DNA toolbox (Montagne et al., 2011; Padirac et al., 2012) is a set of three modules designed to reproduce gene regulation networks dynamics with a simple framework. Those modules, namely activation, autocatalysis and inhibition, use solely DNA strands and enzymes, making both modelization and implementation of systems straightforward (at least when compared to the *in-vivo* lego networks of synthetic biology). DNA sequences have two possible roles: either signal (simply designated as sequences in the following) or templates. The templates are the backbone of DNA toolbox systems, and are used to generate a specific signal

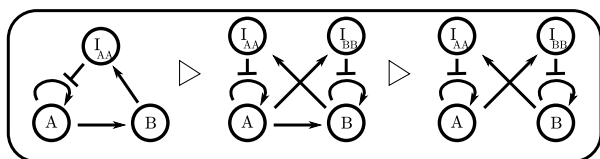


Figure 1: Graphical representation of systems from the DNA toolbox. Nodes represent sequences while arrows represent templates. The Oligator (left) can be mutated into a bistable in two steps. First, an autocatalysis connection  $B$  to  $B$  with an inhibition from  $A$  is added. Then, the activation from  $A$  to  $B$  is removed. Note that those two operations may happen in any order.

from another signal. Specific sequences can also be generated to inhibit a given template. Since they represent the “code”, templates are kept stable over time, and are chemically protected against enzymatic activity that could affect them. Signal sequences, on the other hand, are continuously degraded to keep the system dynamic.

The important feature of the DNA toolbox activatory and inhibitory modules is that they are arbitrarily connectable to each other. The designer of the network freely defines the pattern of interactions by assigning the sequences of the template through Watson-Crick complementarity. For example, a cascade of activation reaction is obtained by mixing a number of bidomain templates such as  $AB$ ,  $BC$  or  $CD$ , where  $A$ ,  $B$ ,  $C$ ,  $D$ , and so on represent orthogonal 11mers. The Oligator from (Montagne et al., 2011), a simple oscillator, is obtained by combining the three templates  $AA$ ,  $AB$  and  $Blaa$  (where  $Iaa$  represent the inhibitor of  $AA$ ). The graph of this system can be seen in Figure 1, left.

One interest of the toolbox in the scope of genetic algorithms is that any modification of the “genome” of an individual (that is, the sequences and templates it is made of, not to be confused with the hypothetical genome their actual DNA strings are encoding) still yields a valid individual (albeit a possibly uninteresting one), and that a wide range of possible behaviors are very few modifications apart. For instance, bioNEAT (see next Section) can jump in two steps from the Oligator (Montagne et al., 2011) to Padirac *et al.*’s bistable system (Padirac et al., 2012), as shown in Figure 1. This helps the algorithm navigating the search space more efficiently, as well as preventing, to some degree, the trap of local optima.

**Individuals and encoding** The individuals we consider are chemical reaction networks playing rock-paper-scissors. Each possible move (rock, paper or scissors) is mapped to a specific chemical species (DNA sequences, more specifically signal sequences from the DNA toolbox). Those species are fixed in advance, so that they are always present. Individuals also have references linking to potential oppo-

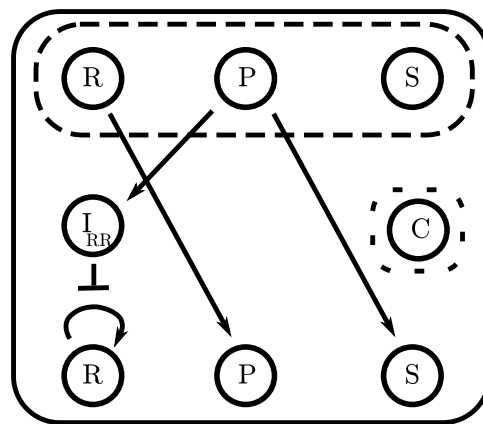


Figure 2: Simple cheating individual displaying both direct and indirect monitoring. Nodes in the dashed box are references to the opponent’s sequence (up) or to the clock (right). By default, this individual will play rock ( $R$ ). If its opponent plays rock or paper ( $P$ ), it will update to play the winning move. Note that this individual does not use the clock.

nents’ corresponding sequences. The main goal of this interface is to allow individuals to react to the opponent’s moves and adapt their strategy over time. Finally, all individuals have a reference to a common clock species, giving them a sense of time. An example of individual is shown in Figure 2.

Individuals are pitted against each other in matches made of ten rounds. The beginning of a round is marked by a spike from the clock sequence. At the end of a round, roughly 20 times the clock’s half-life later, an individual’s move is decided by which of its move sequences has the highest concentration. If the two highest or all such concentration are not different by at least a given threshold, the move is considered invalid, granting the victory to the opponent. Individuals can potentially memorize their opponent’s strategy, since there is no reset between rounds.

**Simulations** The simulation itself was kept simple, with a model similar to that of Padirac *et al.*. In particular, this model doesn’t take into account enzyme saturation. This prevents some advanced strategies (since saturating enzymes may be in itself a way to kill one’s opponent, thus winning by default) and allows individuals to grow without limitations, continuously increasing their size. Since enzymatic saturation creates hidden couplings between the nodes (Rondelez, 2012), removing it was taken as a step to insure the readability of the results. Thanks to this, the behavior of the network - and hence the individual’s strategy - is directly encoded by the networks of cross regulations between the nodes, and not by various type of competitive inhibitions acting at a global level. Using this simplified model is also a compromise between computational requirements and pre-

cision, but any observed behavior should be obtainable in real *in-vitro* experiments.

### bioNEAT: NEAT for Reaction Networks

The evolution of individuals was done by using a modified version of NeuroEvolution of Augmenting Topologies (NEAT) (Stanley and Miikkulainen, 2002), adapted to perform with simulated individual networks built using the DNA toolbox paradigm instead of artificial neural networks. The evolution itself was performed through multiple runs and tweaking of the fitness function.

### NEAT

NEAT is a state-of-the-art evolutionary algorithm designed to evolve both the topology and the parameters of neural networks, while keeping them as simple as possible. This is done by starting from very simple individuals, and progressively complexifying them in a competitive process. This is performed through the addition of new nodes and connections, while at the same time modifying the weight of existing ones.

The major strength of NEAT is that it keeps tracks of when specific connections or node were added in the ancestry line. This allows to perform meaningful cross-over: identical elements present in two individuals, are automatically recognized and matched during the creation of a new individual from two parents. Additionally, mismatching elements from the fittest individual are also passed along.

NEAT also performs speciation to protect innovation that could require more than one step to find a new, better solution to the problem at hand. Specifically, the size of a species depends on the average fitness of its individuals, preventing one type of solution to completely invade the population. Moreover, speciation is easily performed, since the history of evolution of individuals is saved, giving a straightforward distance between individuals based on the genes they possess.

### bioNEAT

Due to the initial resemblance between reaction network and artificial neural network, NEAT stands as a relevant option for optimizing toolbox-based systems. In particular, systems from the DNA toolbox have a straightforward edge/node graph representation similar to neural networks: DNA sequences can be directly mapped to nodes, and connections with positive weights are equivalent to activation links. However, the DNA toolbox cannot be directly implemented using the original NEAT for two reasons. Firstly, additional parameters regarding sequences stability and initial concentration must be added. Secondly, negative links targetting nodes must be replaced by inhibitory links targetting arcs. To address these issues, we introduce bioNEAT, a NEAT-derivative that is able to optimize reaction networks.

A first feature of bioNEAT is to allow the GA to not only modify the “weight” of connections (that is, the concentration of DNA template, in our representation), but also the relevant biological parameters (such as the thermodynamical stability of DNA sequences and their initial concentrations). The thermodynamical parameters of the move sequences was fixed to prevent individuals to use extremely stable sequences to saturate the monitoring of their opponents. In the particular case of the experiments described hereafter, we also prevented activations toward the opponent or the clock.

The second feature of bioNEAT addresses the asymmetry between activation and inhibition process that is inherent to the DNA toolbox, and which cannot be modelled as a classic neural networks link with positive and negative weights. While the sign of a neural weight simply encodes the type of the connection and target a node, a DNA toolbox’ inhibitor targets an *edge* (and impact only one of the output from the source node) rather than a node. Moreover, an inhibitor cannot be instantiated without the template it inhibits. As a consequence, bioNEAT protects the addition of an inhibitory connection (and removal of a particular template) during evolution. Then, bioNEAT produces reaction network with inhibitory connections from node to link.

### Fitness Score

Scoring of an individual uses a lexicographic fitness function taking place in two steps. First, the individual has to beat the three most basic possible players, playing respectively only rock, paper or scissors. This ensures that our individuals are able to play all moves, and to play them discerningly. Individuals unable to pass this test are awarded a very small fitness, based on the number of rounds they have won, directing the evolution toward basic strategies. On the other hand, individuals which were able to pass the test are awarded the right to enter the second phase.

The second phase is a simple tournament among all remaining individuals: each of them has to fight each of the others. The fitness is then based on the amount of correct moves made in total. A sample match is shown in Figure 3. Because of this, the evolutionary pressure forces the individuals into an arms race, to be able to defeat as many opponents as possible.

### Results

Results were obtained by evolving individuals in 10 separate runs, always starting from a uniform population of individuals with autocatalysis on the rock sequence (thus playing always rock). A typical run involved 200 generations of a population of 100 individuals. bioNEAT speciation control loop is adjusted to keep the number of species as close as possible to 10. Other relevant parameters are shown in Table 1. Over the course of the experiment, various kind of strategies emerged before getting outdated or integrated into



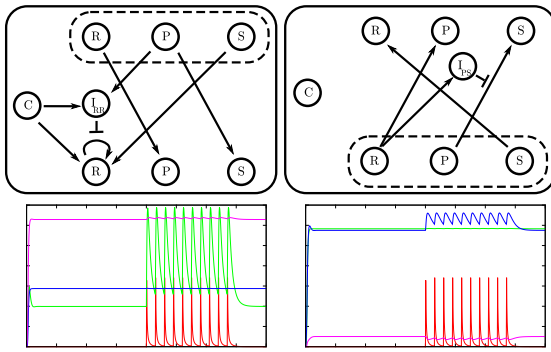


Figure 3: Two fighting individuals. References to the opponent's nodes are shown in the dashed box. Top: the actual network of those individuals. Bottom: the corresponding behavior over time. The color code for sequences concentration is red for the clock, green for rock, blue for paper and purple for scissors. The individual on the right has a better comparison mechanism than the individual on the left, as shown by the fact that it has the correct move before the match starts. However, the individual on the left uses the clock to fake switching his move from scissors to rock, which coerces its opponent to update its move to paper. Just before the round is validated, the individual on the left changes its move again to scissors, winning each hands.

more complex control systems. However, in our runs, a stable group of species typically appeared after 50 to 100 generations and quickly took over the population until the end of the run. They represent individuals which had developed part or all of the mechanisms explained later in this Section, and the apparent stability was only due to a constant arms race, where individuals kept adding more and more modules, while those who couldn't keep up were discarded. However, since our fitness can only compare individuals among a given generation, its evolution over time does not reflect the global improvement of individuals. This prompted us to perform a post-mortem analysis of our individuals by making the best of each generations of a given run fight each other, highlighting a progressive improvement of our individuals, as shown in Figure 4. In particular, the logarithmic shape of the curve goes well with the idea that the efforts required to overcome one's opponents are greater and greater as the simplest strategies get commonly countered.

### Cheating

The easiest, and thus first strategy evolved is actual cheating. Since they have references to what each other will play, and continuous access to current concentrations, the individuals monitor the action of their opponent and try to play accordingly. A minimal example is shown on Figure 2. Cheating can be of two kinds: either using a direct connection ("if my opponent plays rock, I will play paper"), or an inhibition ("if

### General parameters

|                       |     |
|-----------------------|-----|
| Population size       | 100 |
| Number of generations | 200 |

### Speciation parameters

|                               |                          |
|-------------------------------|--------------------------|
| Targeted number of species    | 10                       |
| NEAT compatibility parameters | $c_1 = c_2 = 1; c_3 = 0$ |
| Initial speciation threshold  | 0.6                      |
| Minimal threshold             | 0.1                      |
| Threshold update $\epsilon$   | 0.03                     |

### Mutation parameters

|                                  |      |
|----------------------------------|------|
| P(Mutation only)                 | 0.25 |
| P(Parameter mutation)            | 0.9  |
| Otherwise P(Add node)            | 0.2  |
| Otherwise P(Add activation)      | 0.2  |
| Else add inhibition              |      |
| P(Connection disabling)          | 0.1  |
| P(Gene mutation (for each node)) | 0.8  |

### Crossover parameters

|                           |      |
|---------------------------|------|
| P(Interspecies crossover) | 0.01 |
| P(Re-enabling gene)       | 0.25 |

Table 1: Parameters used to evolve individuals

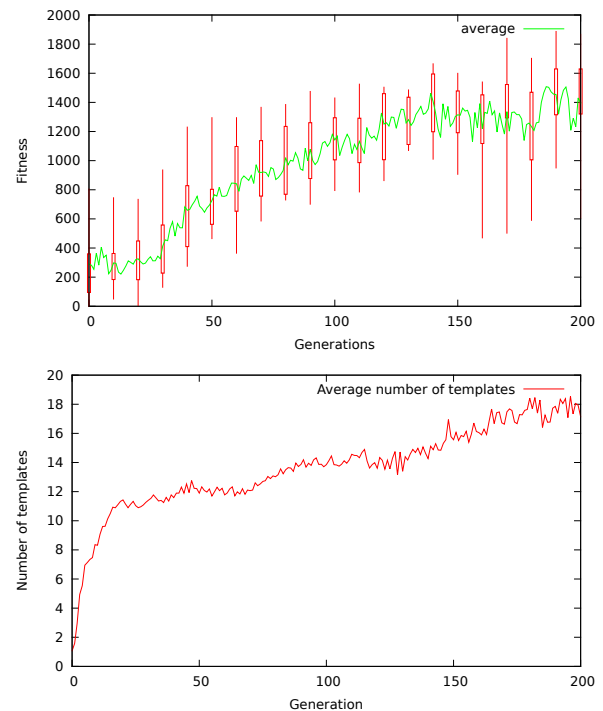


Figure 4: Top: average *a posteriori* fitness of the best individuals, as well as minimum, maximum, first and third quartiles. While noisy, the curve still shows an increasing trend similar to that of a logarithm. Bottom: the average number of templates in individuals over generations in a typical run. The trend is similar to that of the fitness, showing that bloating stays within acceptable limits.

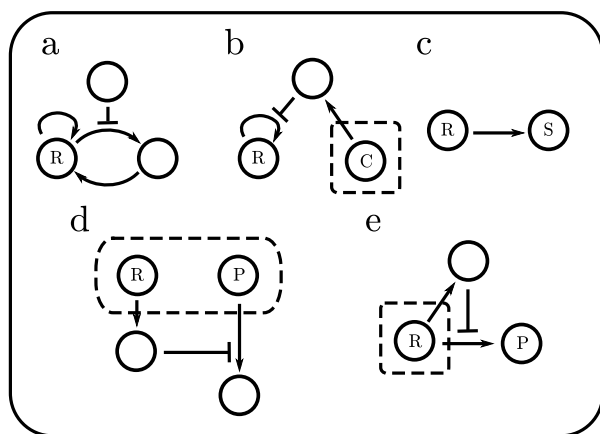


Figure 5: Basic mechanisms observed in individuals. (a.) Noise generation with two activation level. When the additional path is inhibited, the main sequence will still have a high concentration, but not high enough to be this turn's move. (b.) A given move's concentration is kept low for some time by being inhibited by the clock sequence C. (c.) A very simple feint: while pretending to play rock (the sequence R) has a non-zero concentration), the individual is actually playing scissors (S), which would win against the expected reaction of the opponent. This mechanism is often decorated with various other systems to balance the concentrations of one sequence relatively to the other. (d.) Simple comparison mechanism. The reaction path from the opponent's move will only be activated if the concentration of paper (P) is high enough, compared to the concentration of rock (R). (e.) A fold change detector, allowing the monitoring of the increase in the concentration of the rock (R) sequence of the opponent. Often, the detection will happen after a first amplification of the monitored signal.

my opponent plays rock, I will not play scissors"). Cheating leads in some cases to the apparition of oscillatory behaviors, as both individuals are both trying to play the winning move.

### Defense mechanisms

Once cheating appears, it quickly spreads among the whole population, either by cross-over, elimination of individuals which could not adapt, or by parallel discovery of the mechanism. From there on, the only way to improve is to develop mechanisms against the other cheater's spying while at the same time improving the monitoring of its current move. Many defenses were expressed among the evolved individuals, but can mainly be separated into five categories: noise generators, stealth, feint, concentration comparators and fold change detectors. Representatives of all those categories are shown in Figure 5.

Noise generators are the easiest form of defense. Since it is fair to assume that the opponent will monitor at least

two move sequences to decide its own next move, a simple yet efficient way to keep it off track is to continuously generate all sequences. This is a valid action, since only the highest sequence decides which move is played. Having a weak autocatalytic connection is enough, as long as there is a way for the other sequences to become lower (remember that an individual has to be able to play all moves to have a good fitness). Often, such sequence will have an additional catalytic loop using an additional sequence. This loop is only activated when this sequence is supposed to be played. This simple mechanism allows the individual to have multiple activation levels (by opposition to just "on" and "off"), with a better control on the final concentration of the target sequence rather than using activation mechanisms from different possibly not trustworthy part of the system.

Stealth is the complementary of noise generation. Instead of hiding one's true move among decoys, it is kept at a concentration as near to zero as possible until the last moment. This technique relies on monitoring the clock sequence, since timing is extremely important. The clock sequence is used to generate a large amount of timer, which in turn inhibits a specific move. If the inhibition is stable enough, the target sequence will be kept low until the timer has been degraded. If the delay is not long enough, the opponent will still have time to read and adapt. On the other hand, if the delay is too long, the move will not be valid. Part of the system dedicated to this mechanism seems to be very stable over generations, since it is based on a delicate balancing of parameters where any change can prove deadly.

Feint resembles closely the previous two strategies, but uses a different structure. In this case, the individual spoofs a specific move (say "rock"), but this very move also activates the generation of the real move (for instance "scissors"), often through a long activation path to generate delay. It relies on the fact that the opponent will try to adapt to the perceived move, and won't be able to react in time to the change. The system may be reset by the clock, or by a change in the opponent's perceived move.

As the direct monitoring of sequences became less and less reliable, structures to compare absolute concentrations as well as detect sudden modifications became more and more common. Concentration comparison is done through the inhibition of a reaction path if its activation is not strong enough compared to the reference. Since this inhibition originates from the monitoring of another sequence, the first pathway is activated only if the first sequence has a higher concentration. Of course, by tuning the strength of pathways and inhibition, it is possible to have more specific control over the targeted ratio between the two sequences. For instance, it would be possible to slightly modify the system to inhibit the reaction path only if the compared sequence has a concentration multiple times higher than the reference sequence. This defense mechanism is used to counter noise generators and feints.

The last technique commonly spread among individuals is a way to detect concentration increase. While concentration comparison is able to detect that a stealthy move is being played, it is only able to do so once the move became dominant (which, if the other player is timing right, should be too late). However, by using a monitoring coupled with incoherent feedforward, individuals are capable of detecting rapid variations in concentration, which would be a sign that their opponent is about to switch their move. Some individuals also *pretended* to switch their move to throw such defense technique off guard, but this was quickly countered by a mix of both direct comparison and incoherent feedforward.

## Memory vs cheating

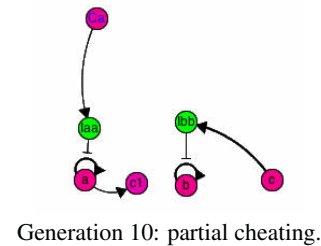
Quite early on, individuals with a basic memory, such as the bistable from Figure 1, appear in the population. However, those individuals were too “naive” in the sense that they had no defense against cheaters. Moreover, cheating requires about the same amount of mutations to appear, or even less if partial (that is, the individual can read *some* moves, but not all). For this reason, it seems that it is much more advantageous for individuals to focus only on attack and defense. This prevented the reappearance of memory in later generation, leading to purely reactive individuals.

## The arms race

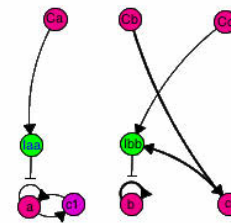
Looking at individuals over time shows the apparitions of the different cheating and defense mechanisms over time, with a noticeable complexification of the best individuals. Figure 6 shows such individuals at different times of a specific run, highlighting the apparition of various mechanisms.

The logical conclusion of this evolution strategy is that individuals with high fitness in a given generation have very little, or even no structures that are not related to cheating and defeating. Even when they exist, such structures are mutated during the next few generations to serve some attack or defense purpose. We performed an *a posteriori* evaluation of the fitness to check whether this increase in individuals size was indeed justified or only bloating. By performing this evaluation, we get a sense of the improvement of individuals over time that cannot be deduced from the lexicographic fitness used for evolution, since the later one only compares individuals from a given generation. The fitness itself is computed by making the best individual of all generations fight each other and score points in the same fashion than in the second part of the lexicographic fitness.

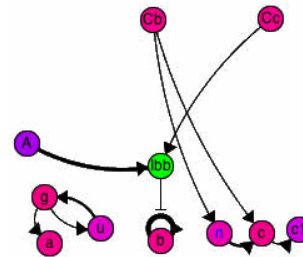
The trend of the *a posteriori* fitness also implies that there is no cyclic effect. While the lexicographic fitness guarantees that all individuals have the capacity of playing any move given the right conditions, there could be more advanced strategy displaying such cyclic dynamics. For instance, individuals using stealth are beaten by individuals using incoherent feedforwards, which could have been, in



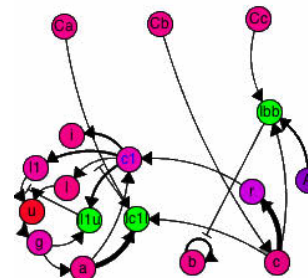
Generation 10: partial cheating.



Generation 14: complete cheater.



Generation 109: stealth. The clock sequence (here designated  $A$ ) hides a move ( $b$ ).



Generation 122: fold change detector. The sequence  $c$  both activates and inhibits the creation of  $a$ . However, the activation path is longer than the inhibition path, meaning that  $a$  (rock) is only activated by this module if the concentration of  $c$  (scissors) is decreasing. Since  $c$  is directly linked to the opponent's  $b$  (paper), this individual is protected against stealthy play of  $b$ .

Figure 6: Individuals generated during a run. The color of activation nodes indicates their stability, going from red (very unstable) to blue (very stable). Green nodes are inhibitors. The notation for the moves rock, paper and scissors is respectively  $a$ ,  $b$  and  $c$ . References to the opponent's sequences are designated by a leading  $C$ .  $A$  represents the clock.

turn, beaten by another strategy that is weak against stealth. Since the fitness increase is monotonic (if we ignore the noise), we can conclude that the arms race is open-ended, with complexification of individuals the only possible way to improve.

We could also note that the arms race pushes individuals to perform well within their own ecosystem, but not always optimally. For instance, the individual from generation 122 in Figure 6 only defends against stealthy changes in the concentration of “paper”, leaving it open to the exact same strategy, if performed on another move. However, it is easy for a human designer to take inspiration from those modules to create an “optimal” player.

## Conclusion

In this work, our first hope was to observe the emergence of memory to allow non-trivial strategies at rock-paper-scissor using bioNEAT, a modified version of NEAT designed to evolve chemical reaction networks from the DNA toolbox. However, the very rules, derived from experimental settings, we set for the games prevented this mechanism from being efficient. Instead, increasingly complex cheating seemed to be the best answer. However, this is not the only thing we learned from this exercise. While having DNA systems compete against each other and evolve new (cheating) strategies can be a goal in itself, the systems evolved along the way gave us also more insight about DNA computing systems. In particular, it was possible to observe the emergence of particular structures with interesting dynamics, which may prove useful to a human trying to develop DNA systems, like with the libraries of (Rodrigo et al., 2011). It could be also interesting to make individuals compete against a human designed “optimal” cheater and see if they can evolve even more advanced strategies to counter it. Furthermore, since the DNA toolbox mimic the behavior of gene regulatory circuits (Montagne et al., 2011), an open question would be whether those mechanisms appear in real life or if they are only valid in the toolbox. Also, it would be interesting to extend the current systems to take into account reaction-diffusion and be able to play more complex games. There is little doubt that such systems will have their own share of remarkable mechanisms.

## References

- Cardelli, L. (2011). Strand algebras for dna computing. *Natural Computing*, 10(1):407–428.
- Cook, R., Bird, G., Lünser, G., Huck, S., and Heyes, C. (2012). Automatic imitation in a strategic context: players of rock–paper–scissors imitate opponents’ gestures. *Proceedings of the Royal Society B: Biological Sciences*, 279(1729):780–786.
- Eiben, A. E. and Smith, J. E. (2003). *Introduction to Evolutionary Computing*. SpringerVerlag.
- Elowitz, M. B. and Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338.
- Frean, M. and Abraham, E. R. (2001). Rock–scissors–paper and the survival of the weakest. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1474):1323–1327.
- Kerr, B., Riley, M. A., Feldman, M. W., and Bohannan, B. J. (2002). Local dispersal promotes biodiversity in a real-life game of rock–paper–scissors. *Nature*, 418(6894):171–174.
- Macdonald, J., Stefanovic, D., and Stojanovic, M. N. (2008). Dna computers for work and play. *Scientific American*, 299(5):84–91.
- Magnasco, M. O. (1997). Chemical kinetics is turing universal. *Physical Review Letters*, 78(6):1190–1193.
- Montagne, K., Plasson, R., Sakai, Y., Fujii, T., and Rondelez, Y. (2011). Programming an in vitro dna oscillator using a molecular networking strategy. *Molecular systems biology*, 7(1).
- Namiki, A., Imai, Y., Ishikawa, M., and Kaneko, M. (2003). Development of a high-speed multifingered hand system and its application to catching. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 3, pages 2666–2671. IEEE.
- Padirac, A., Fujii, T., and Y., R. (2012). Bottom-up construction of in vitro switchable memories. *Proceedings of the National Academy of Sciences*, 109(47):E3212–E3220.
- Purnick, P. E. and Weiss, R. (2009). The second wave of synthetic biology: from modules to systems. *Nature Reviews Molecular Cell Biology*, 10(6):410–422.
- Qian, L. and Winfree, E. (2011). Scaling up digital circuit computation with dna strand displacement cascades. *Science*, 332(6034):1196–1201.
- Reichenbach, T., Mobilia, M., and Frey, E. (2007). Mobility promotes and jeopardizes biodiversity in rock–paper–scissors games. *Nature*, 448(7157):1046–1049.
- Rodrigo, G., Carrera, J., and Jaramillo, A. (2011). Computational design of synthetic regulatory networks from a genetic library to characterize the designability of dynamical behaviors. *Nucleic acids research*, 39(20):e138–e138.
- Rondelez, Y. (2012). Competition for catalytic resources alters biological network dynamics. *Physical Review Letters*, 108(1):018102.
- Seelig, G., Soloveichik, D., Zhang, D. Y., and Winfree, E. (2006). Enzyme-free nucleic acid logic circuits. *science*, 314(5805):1585–1588.
- Sinervo, B. and Lively, C. M. (1996). The rock-paper-scissors game and the evolution of alternative male strategies. *Nature*, 380(6571):240–243.
- Smith, J. M. (1993). *Evolution and the Theory of Games*. Springer US.
- Soloveichik, D., Seelig, G., and Winfree, E. (2010). Dna as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences*, 107(12):5393–5398.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127.



# Barrier Trees for Metabolic Adjustment Landscapes

Christoph Flamm<sup>1</sup>, Chris Hemmingsen<sup>2</sup>, and Daniel Merkle<sup>2</sup>

<sup>1</sup> Institute for Theoretical Chemistry, University of Vienna, Austria

<sup>2</sup> Department for Mathematics and Computer Science, University of Southern Denmark, Denmark

daniel@imada.sdu.dk, xtof@tbi.univie.ac.at

## Abstract

We construct and analyze a discrete fitness landscape called *metabolic adjustment landscape*, from sub-networks covered by different productive flux distributions of a metabolic network. The topological structure of this landscape, i.e., the local minima and saddle points, can be compactly represented as a hierarchical structure called *barrier tree*. The switching from one local optimal flux pattern to another one is accompanied by adjustment costs, since genes have to be turned on or off. This phenomenon gives rise to saddle points in the metabolic adjustment landscape. Our approach allows calculating the minimal cost pathway that connects any two local minima in the landscape. Furthermore, our method yields a detailed ordering which reactions have to be (de-)activated to switch from one flux distribution to another one with minimal adjustment costs. Such a mechanistic hypothesis can guide experimental verification. We apply our approaches to a network describing the central carbon metabolism of *E. coli*.

## Introduction

In recent years information from high-throughput sequencing and metabolomics has been integrated into genome-scale high quality reconstructions of metabolic networks of various organisms (Thiele and Palsson, 2010). These networks are a valuable computational resource in areas such as metabolic engineering, white biotechnology or synthetic biology. In particular questions how changes in the genetic setup of organisms influence the distribution of mass fluxes through a metabolic network have shifted into the focus of research. Flux balance analysis (FBA) (Orth et al., 2010) is among the most popular computational techniques to calculate such flux distributions. Since FBA is an optimization method, the organismal phenotype is usually defined in the form of a biological objective function (Feist and Palsson, 2010) which is optimized under additional constraints that balance or impose bounds on the system.

Another set of fundamental question, very appealing to theorists, focus on how catalyzed reaction networks, i.e. metabolisms, evolve and how novel reaction chemistry emerges during the evolutionary process. In Forst et al. (2006) it was shown that the structure of metabolic networks at the level of the individual chemical reactions, con-

tains enough information for the accurate reconstruction of the phylogenetic relationships. In Flamm et al. (2010) a graph grammar based computational framework for the co-evolution of the enzymes and the metabolic network was described and several scenarios of metabolic network evolution were analyzed. Complexity questions on how to find chemical motifs in chemical reaction networks were analyzed in Andersen et al. (2012). In Schuetz et al. (2012) a large data set of flux distributions measured with <sup>13</sup>C-labeling experiments was analyzed to unveil the principles that govern the distribution and change of metabolic fluxes in *E. coli* under varying conditions. A multi-objective optimization approach together with FBA was used since a combination of several competing objective functions turned out to be best suited for the analysis of the entire data set. Interestingly the study found that flux distribution in *E. coli* is governed by the principle of maximizing production only up to the degree where an easy switch of nutrients is still possible. This behavior seems to be a perfect adaption of *E. coli*'s metabolism to a fluctuating environment where a sudden deficiency of a set of nutrients can be compensated by a switch to other ones accompanied by an easy restructuring of the original flux distribution to the new situation. This comes at the cost that production of compounds (e.g. for building up biomass) in the metabolic network can never be fully maximized to the theoretical limit within the metabolic network of *E. coli*.

While there are several sophisticated computational methods to assign one "optimal" flux distribution to a metabolic network (for a review see Lewis et al. (2012)), to the best of our knowledge, this is the first study that analyzes the entire variety of (optimal) flux distributions over a metabolic network with a fixed genetic setup, but varying activity for subsets of genes in the discrete landscape metaphor.

The brief outline of this paper is as follows: we will first introduce barrier trees for discrete fitness landscapes. Then we will explain different methods how to create metabolic adjustment landscapes, which we use for barrier tree analyzes. We expect the reader to be familiar with the concept of FBA, for an in depth introduction we refer to (Palsson,

2006). In the results section we will first present an artificial example to illustrate our approaches, and then analyze two metabolic adjustment landscapes of *E. coli*.

## Barrier Trees

The switching between different productive flux distributions is accompanied by flux adjustment costs, since genes have to be regulated to achieve the change in the flux distribution. This raises saddle points which connect basins associated with optimal (productive) flux distributions. It seems therefore natural to apply the theory of discrete fitness landscapes to characterize the discrete landscape induced by flux adjustment costs and to get a deeper understanding of its topological and functional structure.

Formally, we define a landscape as a triple  $(X; N; f)$  consisting of a set of configurations  $X$ , a topological structure  $N$  that determines the mutual accessibility of configurations, and a cost or “fitness” function  $f : X \rightarrow \mathbb{R}$ . In our case elements in  $X$  will be metabolic networks. The neighborhood relation  $N$  is typically defined by the “move set”. In this contribution we will restrict ourselves to the simplest case in which the configuration space  $(X; N)$  is a finite directed graph  $G = (X; E)$  with vertex set  $X$  and edge set  $E$ . Here edges connect configurations that can be inter-converted by a single move. (If the move-set is symmetric,  $(X; N)$  can also be represented as an undirected graph.) The fitness value of the lowest saddle point separating two local minima  $x \in X$  and  $y \in X$  is

$$\hat{f}[x, y] = \min_{\mathbf{p} \in \mathbb{P}_{xy}} \max_{z \in \mathbf{p}} f(z) \quad (1)$$

where  $\mathbb{P}_{xy}$  is the set of all paths  $\mathbf{p}$  connecting  $x$  and  $y$  by a series of consecutive operations from the move set.

If the fitness function is non-degenerate, i.e., two configurations have distinct fitness values, then there is a unique saddle point  $s = s(x, y)$  connecting  $x$  and  $y$  characterized by  $f(s) = \hat{f}[x, y]$ . The extension to degenerate fitness functions is discussed in detail in Flamm et al. (2002). To each saddle point  $s$  there is a unique collection of configurations  $B(s)$  that can be reached from  $s$  by a path along which the fitness value never exceeds  $f(s)$ . In other words, the configurations in  $B(s)$  are mutually connected by paths that never go higher than  $f(s)$ . This property warrants calling  $B(s)$  the *valley or basin below the saddle*  $s$ . Furthermore, suppose that  $f(s) < f(s')$ . Then there are two possibilities: if  $s \in B(s')$  then  $B(s) \subseteq B(s')$ , i.e., the basin of  $s$  is a “sub-basin” of  $B(s')$ , or  $s \notin B(s')$  in which case  $B(s) \cap B(s') = \emptyset$ , i.e., the valleys are disjoint. This property arranges the local minima and the saddle points in a unique hierarchical structure which is conveniently represented as a tree, termed *barrier tree*.

## Landscapes of Metabolic Adjustment Networks

A metabolic (reaction) network, is usually represented as a hyper-graph, where the nodes indicate the set of chemical compounds that are connected by hyper-edges corresponding to the set of chemical reactions  $R$ . The power-set of  $R$  induces a whole series of “smaller” instances of metabolic networks, where a subset of reactions is removed from the original metabolic network. A *metabolic adjustment landscape* is a directed graph  $(X, E)$  with a vertex set (or configurations)  $X$ , where each vertex  $x$  corresponds to one of the metabolic network induced by an element of the power-set of  $R$ . The topological structure of the metabolic adjustment landscape is defined by the neighborhood function  $N$ , which determines how the “different” metabolic networks are connected via operations from the move set. In the simplest case the move set is defined by adding or removing exactly one reaction. An edge is labeled with the name of the reaction that has been added or removed (see right of Fig. 1). If the reaction is removed, there is a “-” in front of the name. Two vertices are connected by two edges going in opposite directions, if it is possible to go from one network to the other by adding/removing a reaction. The aforementioned configurations space is converted into a discrete landscape by assigning a fitness value  $f(x)$  to each configuration  $x$ . However, some of the networks are not “viable” in the sense, that these networks cannot support flux between predefined input and output nodes. The viability of a network is decided by running a FBA with a predefined objective function. If the network is viable the assigned fitness value is the number of reactions in that network and infinity otherwise. The rational behind this fitness function is, that the expression of genes, to provide the corresponding chemical transformation in the network in the form of enzymes, is a costly process. In that sense the chosen fitness function quantifies the active or expressed portion of the genetic setup i.e. all possible enzymes encoded in the genome of an organism. More formally

$$f(x) = \begin{cases} |x| & \text{if network } x \text{ is viable} \\ \infty & \text{otherwise,} \end{cases} \quad (2)$$

where  $|x|$  is the number of reactions (or hyper-edges) in a metabolic network  $x \in X$ .

## Unrestricted and Restricted Landscapes

Furthermore, we distinguish two cases of metabolic adjustment landscapes. First, the *unrestricted* case, where any active reaction can be removed and any inactive reaction may be added. In other words two networks are connected if their symmetric difference consists of exactly one reaction. Since any reaction can be removed or added networks can be generated which cannot support flux between source and sink nodes. Hence, unviable networks are valid configurations in unrestricted metabolic adjustment landscapes. Second,

the *restricted* case, where adding and removing reactions is constrained by the following rationales: (i) reactions which would cut the last connection between source and sink nodes cannot be removed (this guarantees, that every network is viable) (ii) reactions can be inserted if all their reactants are produced by “other” reactions already in the network (iii) reactions can be removed if they do not disable following reactions. A reaction is disabled if no other reaction produces its reactants.

### Fitness Functions for Networks in Metabolic Adjustment Landscapes

The standard fitness functions used for FBA are usually defined via a deviation to a given target flux, aiming at biomass production, or aiming at maximizing the production of one or more products. For a critical assessment of the assumptions made in FBA see Schuster et al. (2008). All these fitness functions do not account for the fact, that adding a reaction to a metabolic network induces costs (i.e., the genetic setup needs to be more complicated, as the corresponding enzymes need to be available). More formally this can be phrased as a Lemma.

**Lemma 1.** *Within the landscape of a metabolic adjustment network, any of the standard fitness functions for FBA (Schuetz et al., 2007) leads to a barrier tree with exactly one leaf node.*

*Proof.* A landscape of metabolic networks has a barrier tree with exactly one leaf node (i.e., a barrier tree without any barriers) iff the landscape is (possibly weakly) unimodal. To show the unimodality, consider a flux balance analysis on a metabolic network  $x$  that reaches an optimal fitness score  $f(x, v)$  with a certain optimal flux distribution  $v$  (the flux vector  $v$  assigns fluxes to all reactions). The core observation to be made is, that extending network  $x$  to a network  $x'$  can not lead to a worse fitness score, as the flux induced by  $v$  in the extended network reaches at least the fitness score  $f(x, v)$  in the network  $x'$ . Adding all possible reactions successively will therefore always lead to the best possible fitness score (without the need to cross a barrier) and two strict local minima can not exist in the metabolic adjustment landscape.  $\square$

Based on the above observation that no barriers can appear if a standard fitness functions for a FBA is used to analyze metabolic adjustment landscape, we use as a natural motivation in creating a new fitness function, that adding a reaction needs to be penalized, i.e., the genetic setup is more complicated. We therefore defined the fitness function as given in Eqn. 2. Note, that this ensures that extending a metabolic network by a reaction leads to a worse fitness score, even if the added reaction is not used in an optimal flux in the extended network (where optimal is meant wrt. to the objective function used for the FBA).

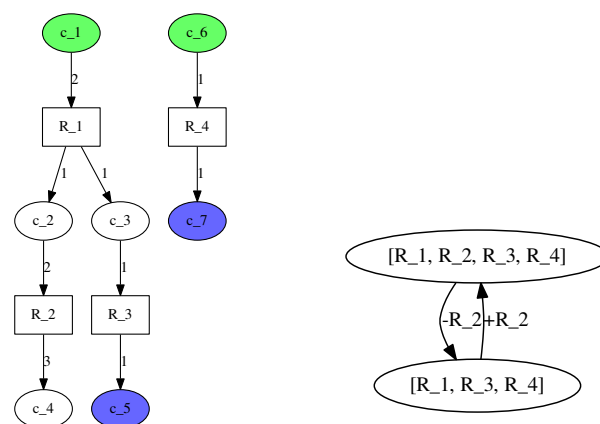


Figure 1: Toy example for a restricted metabolic adjustment landscape: The four given reactions are  $R_1, \dots, R_4$ , educts are  $c_1$  and  $c_6$  (green circles), and the target compounds are  $c_5$  and  $c_7$  (blue circles). Left: the metabolic network consisting of all reactions, the stoichiometric coefficients of the reactions are given by the weights on the hyperedges. Right: the resulting restricted metabolic adjustment landscape.

Fig. 1 shows an illustrative example with four reactions  $\mathcal{R} := \{R_1, \dots, R_4\}$ , the set of educts is  $\{c_1, c_6\}$  (sources), and the set of products is  $\{c_5, c_7\}$  (sinks). The restricted landscape has two nodes only, the fitness values of the two metabolic networks are 3 (in the network without reaction  $R_2$ ) and 4 (in the network with  $R_2$ ). The unrestricted landscape consists of  $2^{|\mathcal{R}|} = 16$  metabolic networks. The additional 14 metabolic networks are all unviable, since removal of reactions  $R_1, R_3$  or  $R_4$  disconnects the last path between the source and the sink nodes, rendering a productive flux impossible. Therefore the fitness value of these networks is set to  $\infty$ .

## Results

In the first part we will present results for an artificial example to illustrate our approach to infer a barrier tree from a metabolic adjustment landscape. In the second part we will apply our approach to the metabolic adjustment landscape derived from the main reactions in the central carbon metabolism of *E. coli*.

### Artificial Example

We used a network of 23 compounds and 25 reactions as depicted in Fig. 2. Compound  $c_{17}$  is the only substrate (source) and compound  $c_5$  is the only product (sink) compound to be produced in a quantity of 0.1. We computed the restricted metabolic adjustment landscape, which consists of 561 metabolic networks. This is relatively small in contrast to the unrestricted landscape which is comprised of  $2^{25}$  networks, the majority of these networks are unviable. The restricted landscape was analyzed: it has 5 local min-

ima, all of them reach a fitness value of 10 reactions. Note, that any local minima can not have an unused reaction in an optimal flux distribution, as this reaction could be removed (leading to a better fitness in the metabolic adjustment landscape) while keeping the same fitness for the FBA of the reduced network. The barrier tree is shown in Fig. 3, the metabolic networks that correspond to minima 1 and 2 are depicted in Fig. 2 (the red nodes and edges, are only shown for illustration purposes and are not part of the solution networks which correspond to the local minima 1 and 2, these parts are shown in green). The barrier tree calculation allows backtracking the minimum cost path between any two minima in the barrier tree. Such a minimum cost path leading from minimum 1 to minimum 2 on the restricted metabolic adjustment landscape would (i) add the reactions with indices 10, 13, and 15, (ii) remove the reactions with indices 7, 8, and 12 (this became possible since step (i) introduced a parallel path connecting the source to the sink node keeping the network viable). This minimum cost path results in a barrier of 13 (denoted as B1 in Fig. 3) between the two local minima 1 and 2 since the saddle point network contains exactly 13 active reactions. The relations between the local minima and the minimum saddle points in the metabolic adjustment landscape can compactly be represented as a barrier tree Fig. 3. The internal tree nodes are the minimum saddle points between local minima which are located at the leaf nodes. For a change to a flux pattern that uses only the pathway via reaction indices 20, 19, ..., a barrier of height 20 needs to be crossed. This barrier is denoted as B2 in Fig. 3.

### The Central Carbon Metabolism as a Restricted Landscape

The Central Carbon Metabolism (CCM) is a union of well known catabolic pathways, such as glycolysis and tricarboxylic acid cycle (TCA), and a minimal number of “interface reactions” to important anabolic pathways, found in all three kingdoms of life. The representation of the network we use originates from de Figueiredo et al. (2008). Their network has 37 reactions, provided that both directions of reversible reactions are counted separately. The target compound is glucose-6-phosphate (G6P). The network can “feed” on different substrates (source compounds) to achieve the production of G6P (target compound). Among them are Acetyl coenzyme A (AcCoA) created as degradation product of the fatty acids metabolism, the two amino acids Alanine (Ala) and Aspartic acid (Asp), or Pyruvate (Pyr), the simplest  $\alpha$ -keto-acid derived as end-product of Glycolysis. We pruned unused reactions from the original network to increase computational speed. For this an elementary mode (EM) analysis (Papin et al., 2004) of the CCM with and without Isocitrate lyase EC 4.1.3.1 (ICL) and the Malate-Aspartate shuttle (MAS) using each substrate was conducted. (An elementary mode is a certain feasible flux distribution; all elementary modes can be created by com-

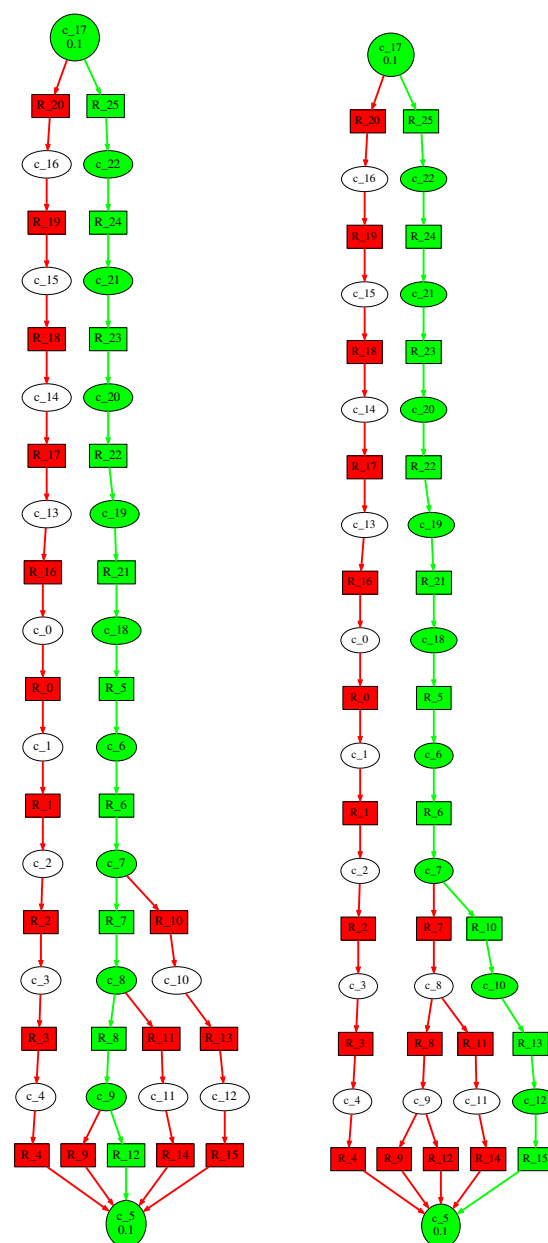


Figure 2: Minima 1 (left) and 2 (right) of the artificial example. Target is  $c_{5\_cell}$  with an outflux of 0.1;  $c_{17\_cell}$  is a substrate with an influx of 0.1; the weights on the edges indicate a steady-state flux distribution. Note that the red nodes are not in the corresponding metabolic network, but are only depicted for illustration reasons.

binning the extreme pathways (EPs) of the network, which are formally derived from the basis vectors of the null-space of the stoichiometric matrix of the network. The EPs are therefore a subset of the elementary modes.) From the set of all possible EMs it can be seen that some directions of reversible reactions are never used. We also removed the linear pathway for creation of “G6P” from Fructose-



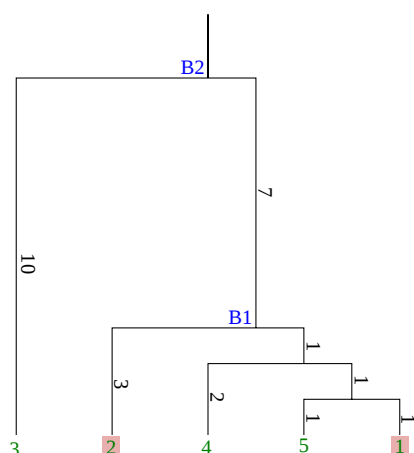


Figure 3: Barrier tree for the metabolic adjustment landscape of the artificial example: minima 1 and 2 (both reaching objective value 10, cmp. Fig. 2) are connected via the lowest saddle B1, which has an objective value of 13. The height difference of nodes (i.e. minima and saddles) in the barrier tree corresponds to the difference wrt. their objective values. The barrier B2 has objective value 20.

1,6-bisphosphate (F1,6PP) using the 3 enzymes Fructose-bisphosphatase EC 3.1.3.11 (FBP1), Phosphofructokinase EC 2.7.1.11 (PFKL) and Glucose-6-phosphate-isomerase EC 5.3.1.9 (PGI), as any solution capable of producing "G6P" would use "F1,6PP" and this pathway. After pruning the network contains 22 reactions, having "F1,6PP" as target compound. The pruned network is depicted in Fig. 7.

**Barrier Tree** We use our approaches to determine the restricted metabolic adjustment landscape for the network given in Fig. 7. For the FBA "F1,6PP" served as target compound and AcCoA, Ala, Asp and Pyr as individual source compounds. We used the tool FASIMU (Hoppe et al., 2011) to convert the reaction networks into the integer linear programs (ILPs), which were solved using IBM's commercial program IBM ILOG CPLEX (2012) (currently freely available for academic purposes). It took 2.2 seconds to build the restricted metabolic adjustment landscape, which contains 12853 vertices. It took 36 minutes for FASIMU and CPLEX to formulate and run all 12853 simulations, i.e. on average 5.95 simulations per second. The resulting barrier tree can be seen as Fig. 4. The barrier tree has 8 local minima, the flux distributions of minima 1, 7, and 8 are depicted in Fig. 5.

**Minima** Although the biological discussion of the results is out of the scope of this paper, it should be noted that the barrier tree nicely illustrates the shift from using the glyoxylate shunt in CCM (minima 8 (cmp. Fig. 5) and 6 (not depicted)) towards not using it (all other minima).

**Barriers** The leaf nodes of the barrier tree show the CCMs different abilities to produce "F1,6PP". The barriers are

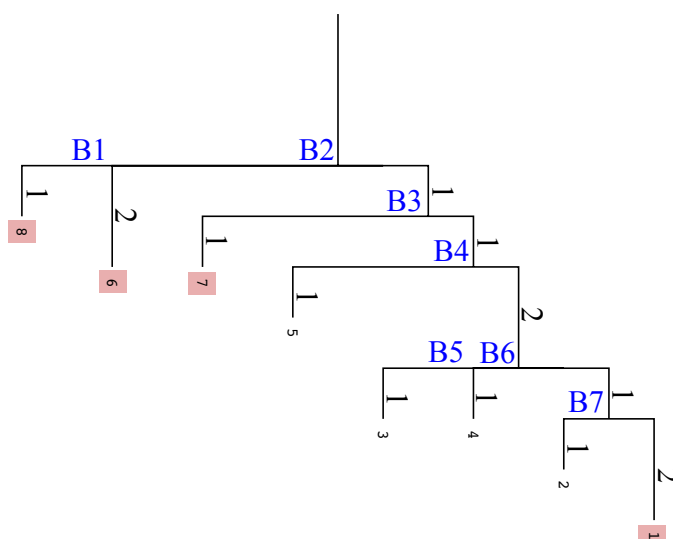


Figure 4: Barrier tree from the restricted metabolic adjustment landscape of CCM for "F1,6PP" production as objective function. Barriers are marked B1 to B7. Min 1: Use Asp. Min 2: Use Ala. Min 4: Use Asp, identical to Min 3, but with ME1\_1 and ME1\_2 blocking each other from being removed. Min 3: Use Ala. Min 5: Use Glu (utilize pathway from OG to MAL). Min 7: Use Glu (utilize pathway from OG to MAL). Min 6: Utilize AcCoA via ICL/MAS straight to OAA from Mal. Min 8: Utilize AcCoA via ICL/MAS detour over ME1\_1 and PC from Mal to OAA.

marked B1 to B7. Their fitness indicates the least amount of reactions a network needs to have when passing a barrier that is necessary to connect two metabolic networks.

Let  $M_7$  (resp.  $M_8$ ) be the network that minima 7 (resp. 8) represent. Imagine we want to modify  $M_7$  such that it becomes  $M_8$ , while always being able to maintain "F1,6PP" production and minimizing the maximal fitness of all intermediate networks. The computed minimal cost path is as follows:

|    |                                 |                  |           |
|----|---------------------------------|------------------|-----------|
| 1  |                                 | <b>Minima</b> 7  | Score: 11 |
| 2  | Add GPT_1                       | <b>Saddle</b> B3 | Score: 12 |
| 3  | Remove ME1_1                    |                  | Score: 11 |
| 4  | Remove FH_1                     |                  | Score: 10 |
| 5  | Remove SDHA, SDHB, SDHC, SDHD_1 |                  | Score: 9  |
| 6  | Remove SUCLG2, SUCLG1, SUCLA2_1 |                  | Score: 8  |
| 7  | Remove OGDH, DLST, DLD          | <b>Minima</b> 2  | Score: 7  |
| 8  | Add GOT1_1                      | <b>Saddle</b> B7 | Score: 8  |
| 9  | Remove PC                       |                  | Score: 7  |
| 10 | Remove GPT_1                    | <b>Minima</b> 1  | Score: 6  |
| 11 | Add CS                          |                  | Score: 7  |
| 12 | Add ACO1, ACO2_1                |                  | Score: 8  |
| 13 | Add ICL                         |                  | Score: 9  |
| 14 | Add SDHA, SDHB, SDHC, SDHD_1    |                  | Score: 10 |
| 15 | Add FH_1                        |                  | Score: 11 |
| 16 | Add MDH1, MDH2_1                |                  | Score: 12 |
| 17 | Add MAS                         | <b>Saddle</b> B2 | Score: 13 |
| 18 | Remove GOT1_1                   |                  | Score: 12 |
| 19 | Remove GLUD1_2                  | <b>Minima</b> 6  | Score: 11 |
| 20 | Add ME1_1                       |                  | Score: 12 |
| 21 | Add PC                          | <b>Saddle</b> B1 | Score: 13 |
| 22 | Remove MDH1, MDH2_1             | <b>Minima</b> 8  | Score: 12 |

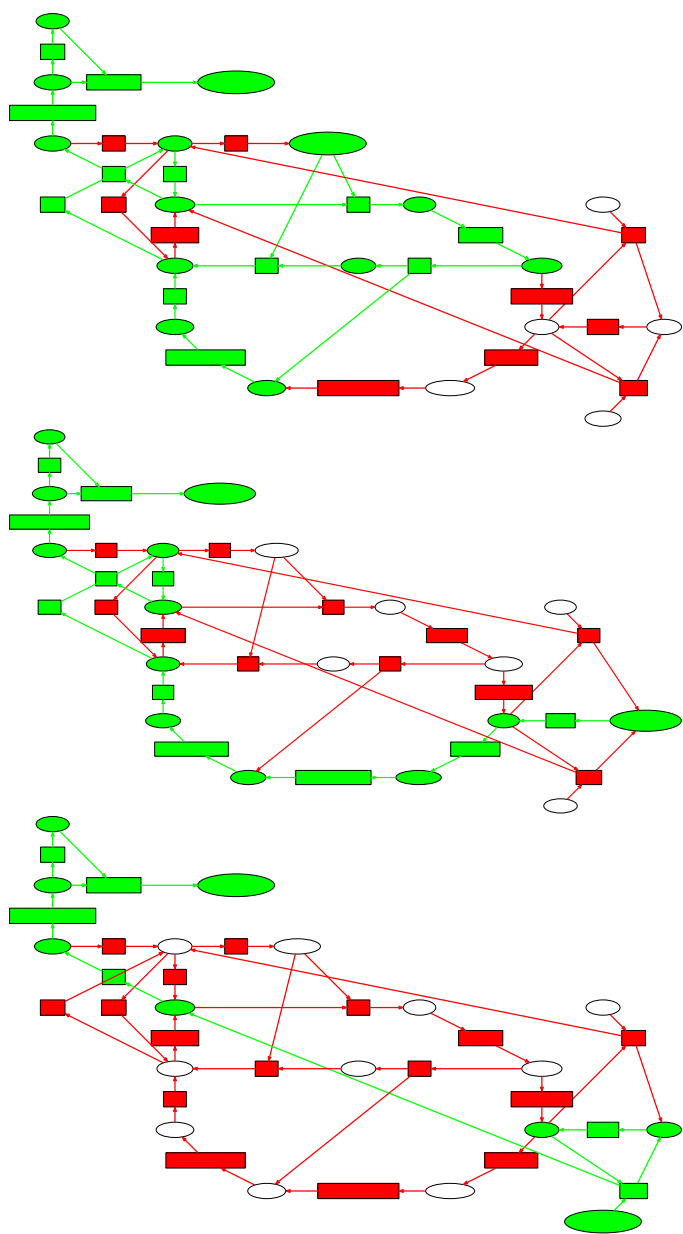


Figure 5: Minima 8 (top), 7 (middle), and 1 (bottom). Note, that the red nodes are not in the corresponding metabolic network, but are only depicted for illustration reasons.

The path never goes above 13 reactions. Although, based on the “closeness” of minima 7 and 8 in the barrier tree, one might think that connecting the two minima while not using more than 13 reactions would result in a short path, the example shows this is in general not trivial. It takes 22 steps of adding and removing a reaction. First “GPT\_1” is added to enable a shift of substrate from “Glu” to “Ala”. Using “Ala” means the 4 reactions along the path from “OG” to “OAA” are no longer needed. These are removed, as well as “ME\_1”, which is not needed either. Now “GOTH1\_1” is added to switch from using substrate “Ala” to “Asp”. This

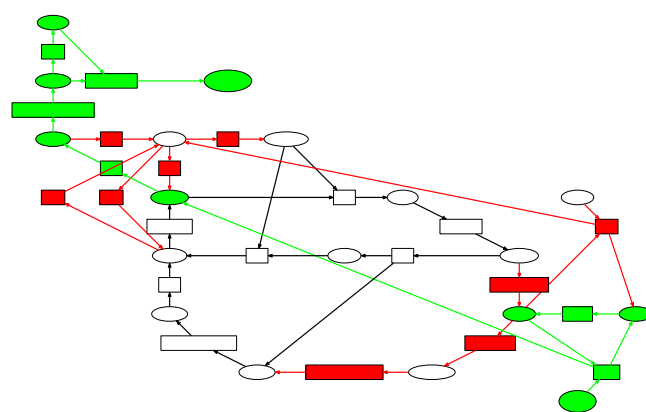


Figure 6: Barrier B2. Note, that the red nodes are not in the corresponding metabolic network, but are only depicted for illustration reasons. The black lines depict reactions that *are* in the metabolic network but are not used in the optimal flux distribution.

removes the need for “PC” and “GPT\_1”. Now all the reactions part of the glyoxylate cycle are added, 2 of which had been removed just some steps ago. Now a shift from using “Asp” to using “AcCoA” as a substrate can be made. This transition state is barrier “B2” and is shown in Fig. 6. “GOT1\_2” and “GLUT1\_2” is now no longer needed. Last thing needed is to use “ME1\_1” and “PC” to reach “OAA” from “Mal”, instead of using “MDH1,MDH2\_1”.

### The Shift from Using Fatty Acids to Amino Acids as an Unrestricted Landscape

This section presents results that illustrate how our approach is used in order to analyze the shift between different given fluxes. Using this method instead of finding all sub-networks reduces the number of simulations, but also requires some sensible choices for the two networks. Here, we choose the networks such that both networks are subsets of the CCM and produce “F1,6PP”, but one does it using fatty acids (“AcCoA”), and the other one does it using amino acids (“Ala”, “Glu”, or “Asp”). For a biochemical discussion on the usage of fatty acids or amino acids to produce glucose see de Figueiredo et al. (2008). Both networks are shown in Fig. 7. Note, that in contrast to the restricted case, we disallow here the removal of reactions that appear in both networks (depicted as yellow hyper-edges in Fig. 7). We use the unrestricted transformation method in order to analyze the metabolic adjustment landscape. To transform the base networks topology to the target networks, 6 reactions must be removed and 6 reactions must be added. It took 0.6 seconds to build the resulting landscape, which contains  $2^{12} = 4096$  vertices. It took 10 minutes for FASIMU and CPLEX to formulate and run all 4096 simulations (i.e., on average 6,83 simulations per second).

**Barrier Tree** The barrier tree for the metabolic adjustment landscape is depicted in Fig. 8. The tree has 4 minima that

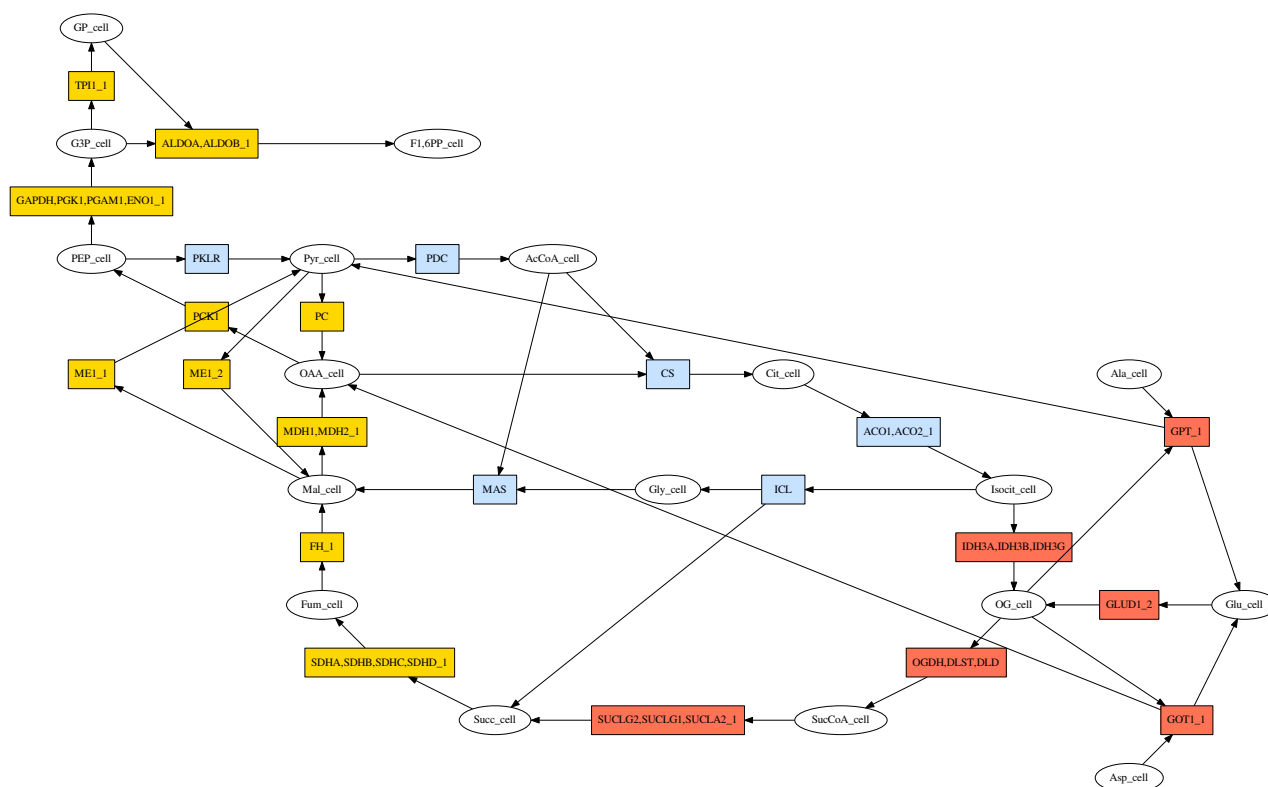


Figure 7: Two networks with an overlap that use different source compounds to analyze substrate switching: colored in yellow are reactions common to both networks, colored in blue are the additional reactions specific for the network utilizing fatty acids, and reactions colored in red are specific for the network utilizing amino acids.

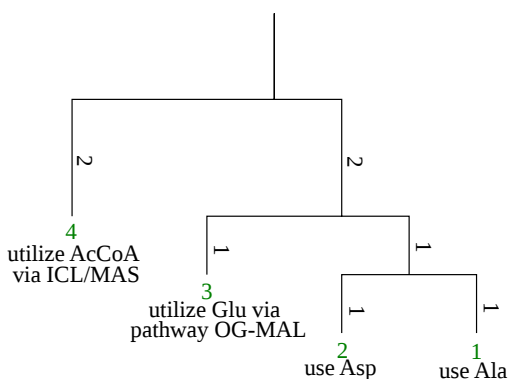


Figure 8: Barrier tree illustrating the shift from utilizing amino acids (minima 1, 2, and 3) to utilizing fatty acids (minimum 4).

nicely illustrates the usage of amino acids (minima 1-3) or fatty acids (minimum 4) as source compounds to produce the target compound "F1,6PP". While changing the flux pattern between different amino acids requires passing only a small barrier of 1-2 additional reactions, switching to a flux pattern that utilizes fatty acids, in contrast, requires to pass a rather high barrier of additional 4 reactions.

**Minima** 2 of the four minima are depicted in Fig. 9. Reactions that have been removed from the network are marked

red, and the flux distribution is shown in green. Each minimum uses different substrates. Minima 1 and 2 show optimal use of both "Ala" and "Asp". They both avoid using the metabolic pathway from "OG" to "MAL", and also do not use "ME1.2". The reason for the large amount of unused "black" reactions is, that they are not allowed to be removed since they are present in the intersection of the two networks. Minima 3 uses "Glu" as source compound and requires the usage of part of the TCA cycle (from "OG" to "MAL") to connect to the target compound "F1,6PP". The barrier tree again suggests that using "AcCoA" as the only substrate requires more active enzymes than in the case of "Ala" or "Asp". In other words the amino acids are a "much cheaper" resource to produce "F1,6PP" from then "AcCoA".

## Conclusions

We introduced a systematic approach to characterize the flux landscapes of a metabolic network. The genetic setup of the metabolic networks is the always same (forming a super-network); embedded are different optimal flux distributions for varying substrate usage and/or target compound productions. Switching between different flux distributions is accompanied by adjustment costs since inactive genes have to be activated and active genes have to be deactivated.

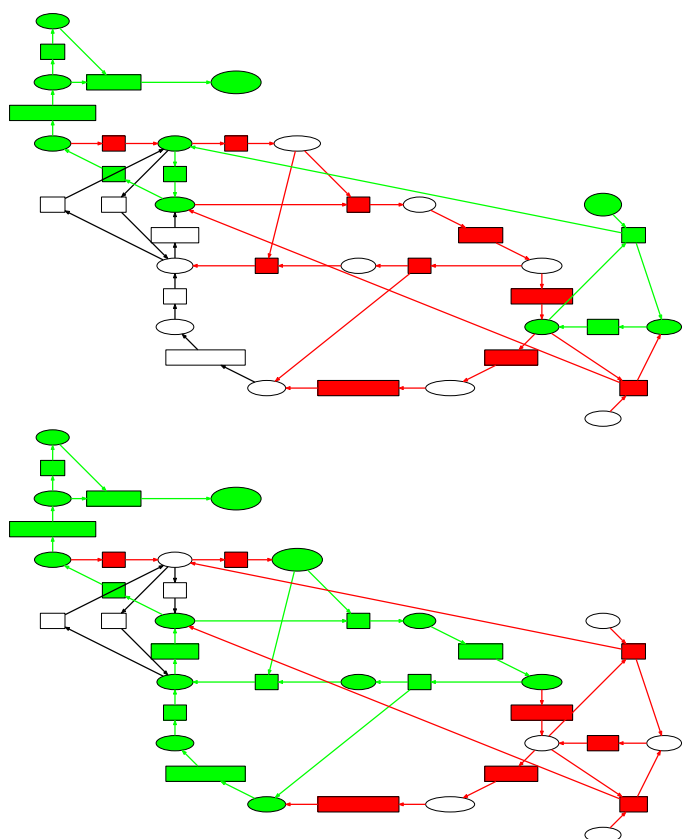


Figure 9: Minima 1 (top): Only uses substrate "Ala" in an optimal fashion by avoiding the metabolic pathway from "OG" to "MAL". Minima 4 (bottom): Only uses AcCoA as substrate and utilizes the glyoxylate shunt by activating the reactions "ICL" and "MAS". Note, that the red nodes are not in the corresponding metabolic network, but are only depicted for illustration reasons. Black edges are unused existing edges in the optimal flux distribution.

From the networks induced by the subset of active reactions (genes) a discrete landscape can be constructed, which we termed *metabolic adjustment landscape*. This landscape is analyzed in terms of local minima and connecting saddle points, and can be efficiently visualized in a hierarchical structure called *barrier tree*. The analysis allows us to find the cost for changing from one optimal flux pattern to another. Furthermore, for the first time, we can calculate in mechanistic detail how this minimal cost pathway looks like, in particular in which order the reactions have to be (de-)activated to achieve the change in the flux distribution. This mechanistic hypothesis can be tested by experimental approaches.

## Acknowledgments

This work was supported by the Danish Council for Independent Research, Natural Sciences.

## References

Andersen, J. L., Flamm, C., Merkle, D., and Stadler, P. F. (2012). Maximizing output and recognizing autocatalysis in chemical

reaction networks is NP-complete. *J Sys Chem*, 3(1).

de Figueiredo, L. F., Schuster, S., Kaleta, C., and Fell, D. A. (2008). Can sugars be produced from fatty acids? A test case for pathway analysis tools. *Bioinformatics*, 24(22):2615–2621.

Feist, A. M. and Palsson, B. Ø. (2010). The biomass objective function. *Curr Opin Microbiology*, 13:344–349.

Flamm, C., Hofacker, I. L., Stadler, P. F., and Wolfinger, M. T. (2002). Barrier trees of degenerate landscapes. *Z Phys Chem*, 216:155–173.

Flamm, C., Ullrich, A., Ekker, H., Mann, M., Högerl, D., Rohrschneider, M., Sauer, S., Scheuermann, G., Klemm, K., Hofacker, I. L., and Stadler, P. F. (2010). Evolution of metabolic networks: a computational frame-work. *J Systems Chem*, 1(4).

Forst, C. V., Flamm, C., Hofacker, I. L., and Stadler, P. F. (2006). Algebraic comparison of metabolic networks, phylogenetic inference, and metabolic innovation. *BMC Bioinformatics*, 7(67).

Hoppe, A., Hoffmann, S., Gerasch, A., Gille, C., and Holzhütter, H.-G. (2011). FASIMU: flexible software for flux-balance computation series in large metabolic networks. *BMC Bioinformatics*, 12:28.

IBM ILOG CPLEX (2012). High-performance mathematical programming engine. v12.5, <http://www.ibm.com/software/integration/optimization/cplex>.

Lewis, N. E., Nagarajan, H., and Palsson, B. Ø. (2012). Constraining the metabolic genotype-phenotype relationship using a phylogeny of *in silico* methods. *Nature Reviews Microbiology*, 10:291–305.

Orth, J. D., Thiele, I., and Palsson, B. Ø. (2010). What is flux balance analysis? *Nature Biotechnology*, 28(3):245–248.

Palsson, B. (2006). *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, first edition.

Papin, J. A., Stelling, J., Price, N. D., Klamt, S., Schuster, S., and Palsson, B. Ø. (2004). Comparison of network-based pathway analysis methods. *Trends Biotechnol.*, 22(8):400–405.

Schuetz, R., Kuepfer, L., and Sauer, U. (2007). Systematic evaluation of objective functions for predicting intracellular fluxes in *Escherichia coli*. *Molecular Systems Biology*, 3:119.

Schuetz, R., Zamboni, N., Zampieri, M., Heinemann, M., and Sauer, U. (2012). Multidimensional optimality of microbial metabolism. *Science*, 336(6081):601–604.

Schuster, S., Pfeiffer, T., and Fell, D. A. (2008). Is maximization of molar yield in metabolic networks favoured by evolution? *J Theor Biology*, 252:497–504.

Thiele, I. and Palsson, B. Ø. (2010). A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nature Protocols*, 5:93–121.



# Reconstitution of the protein synthesis system on a glass microchip

Yoshihiro Shimizu<sup>1</sup> and Yo Tanaka<sup>1</sup>

<sup>1</sup>RIKEN Quantitative Biology Center (QBiC), Japan  
yshimizu@riken.jp

## Abstract

The field of synthetic biology aims to understand the mechanisms of biological systems by designing and constructing artificial biological systems from molecular parts. One of the ultimate goals of the approach is to develop an artificial cell that is under the control of scientists. Although the creation of a bacterial cell controlled by a chemically synthesized genome was accomplished in recent years (1), the construction of a self-replicable artificial cell by organizing essential purified biological macromolecules remains to be achieved (2). Such bottom-up approach is an alternative way for the construction of minimal cells to the top-down approach utilizing natural living cells, which can be helpful for us to explore the boundary between life and non-life.

Here, we reconstituted the protein synthesis system on a glass microchip. By integrating the PURE (Protein synthesis Using Recombinant Elements) system (3), a reconstituted cell-free protein synthesis system composed of purified factors and enzymes responsible for the gene expression in *Escherichia coli*, on a glass microchip, we performed the GFP synthesis with continuous flow in a microchip for the prototype verification.

Biotinylated DNA template encoding the gene for GFP was immobilized on streptavidin-coated sepharose beads with diameter of 34  $\mu\text{m}$ . Prepared beads were introduced into a Y-shaped microchannel in a glass microchip with a 10- $\mu\text{m}$  height dam structure fabricated by 2-step HF wet etching method (Fig. 1) (4). By introducing the components of the

PURE system into this microchip by syringe pump, fluorescent intensity of the recovered solution was demonstrated to be higher than the solution before introduction, indicating that the GFP was successfully synthesized on the microchip.

Because the device has an ability to comprise DNA molecule within it, which can be transcribed to RNA or translated into the protein, the results indicate that it has a potential to be a container of reconstituted life with genetic information. Furthermore, all the macromolecules of the PURE system, tRNA, ribosome, translation factors, RNA polymerase, and several enzymes, consist of RNA and proteins, which can be synthesized by the PURE system, the study sheds light on the way of reconstituting self-replication inside the device, which is crucial for the construction of artificial cells.

- (1) Gibson, D. G. *et al.*, (2010) Creation of a bacterial cell controlled by a chemically synthesized genome. *Science* 329: 52-56.
- (2) Forster, A. C. and Church, G. M. (2006) Towards synthesis of a minimal cell. *Mol. Syst. Biol.* 2: 45.
- (3) Shimizu, Y. *et al.*, (2010) Cell-free translation reconstituted with purified components. *Nat. Biotechnol.* 19: 751-755.
- (4) Sato, K. *et al.*, (2004) Microchip-based enzyme-linked immunosorbent assay (microELISA) system with thermal lens detection. *Lab Chip*. 4: 570-575.

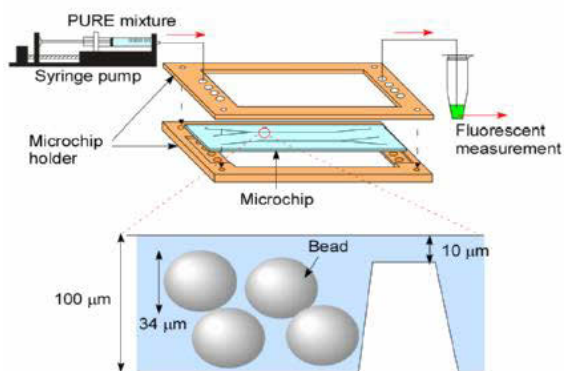


Fig. 1. Design of a microchip

## Recent advancements in synthetic cells research

Pasquale Stano<sup>1</sup>, Erica D'Aguanno<sup>2</sup>, Paolo Carrara<sup>1</sup>, Alfred Fahr<sup>2</sup> and Pier Luigi Luisi<sup>1</sup>

<sup>1</sup>Science Department, University of Roma Tre, Viale G. Marconi 446, 00146 Rome, Italy

<sup>2</sup>Pharmacy Institute, Friedrich-Schiller-Universität Jena, Lessingstrasse 8, 07743 Jena, Germany  
pasquale.stano@uniroma3.it

### Abstract

This paper describes our recent investigations on the construction of synthetic cells. By following a bottom-up synthetic biology approach, we aim at constructing minimal synthetic cells based on the encapsulation of DNA, RNA and proteins within liposomes. We will firstly comment on the physics of solute entrapment inside liposomes, giving emphasis on a remarkable self-concentration effect discovered by us (Luisi et al. 2010, Souza et al. 2011, 2012). Next we will show how it is possible to exploit this phenomenon to reveal the formation of primitive-like, metabolically active cells starting from diluted macromolecular solutions (Stano et al., submitted). In conditions where a protein-synthesis reaction does not proceed at a significant rate, lipid vesicles can entrap all required solutes allowing intraliposome protein production. The second topic deals with the formation of simple, rudimentary primitive cell communities based on giant vesicles (GVs). Oleate-containing GV's associate between each other in the presence of poly-L-arginine to form clusters that might be taken as model of primitive cell communities. Their formation, driven by simple primitive electrostatic interactions bring about a series of distinctive features (stability, enhanced permeability, solute capture, fusion) that might emphasize the role of cooperation in origin of life scenarios, flanking the usual competition issues (Carrara et al., 2012).

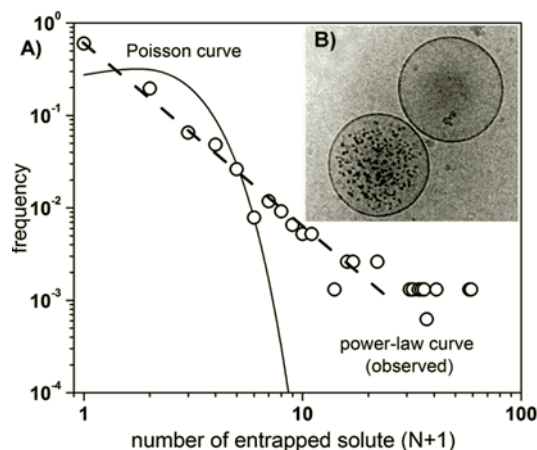
### Synthetic Cells

One of the goals of synthetic biology is the construction of synthetic cells. In addition to the classical synthetic biology approach, based on the manipulation of living organisms, synthetic cells can be constructed by assembling separated molecular parts like lipids, DNA, RNA, proteins in cell-like systems called semi-synthetic minimal cells (Luisi et al., 2006). Born within the community of origin of life, modern semi-synthetic minimal cell research also looks for possible biotechnological applications (but whereas self-organization, spontaneous assembly, and primitiveness are key features of synthetic cell studies in origin of life context, controlled assembly, efficiency and reproducibility are distinctive facets of biotechnology).

Here we will firstly show how a large number of macromolecules self-concentrate inside lipid vesicles (liposomes), bringing about a remarkable rate acceleration of a complex biochemical reaction. Then, we will present our first attempt to build protocellular communities based on the physical association of giant vesicles (GVs).

### Solute entrapment and “super-concentration”

Semi-synthetic minimal cells are defined as those cell-like particles containing the minimal and sufficient number of biological compounds (nucleic acids, protein, lipids) that would allow self-maintenance, self-reproduction, and the possibility to evolve (Luisi et al., 2006). Protein synthesis is a key module of semi-synthetic minimal cell, representing not only a necessary function of the synthetic cell, but also a good model for metabolic complexity. Following our initial observation of protein synthesis in “small” conventional liposomes (200 nm diameter) (Souza et al., 2009) we started a systematic electronmicroscopy study that revealed how macromolecules like ferritin (Luisi et al., 2010), ribosomes (Souza et al., 2011) and ribo-peptidic complexes (Souza et al., 2012) were spontaneously encapsulated within liposomes (Figure 1).



**Figure 1.** (A) Poisson curve versus power-law. (B) ferritin-containing and empty liposomes. Reproduced from Luisi et al. (2010) with permission from Wiley.

Intriguingly, it was found that the intra-liposome solute distribution function does not follow – as expected – the Poisson law, but it is rather shaped as a power law (Figure 1a). In particular, although the vast majority of liposomes encapsulate a limited number of solutes, a very small fraction of liposomes (typically around 0.1%) contains instead a very high number of solutes, so that their internal concentration can

be also one order of magnitude higher than the concentration of the solutes in the environment (Figure 1b). In other words, during liposome formation, solutes can self-concentrate within the liposome cavity. We reasoned that the “super-concentration” effect could drive the formation of functional cells, for instance “protein-producing” ones, even if the low concentration of the solutes in the environment does not allow it. By diluting the transcription-translation kit at a level where no protein synthesis takes place, we simulated a primitive scenario where solutes were already present in diluted form in a sea or fresh-water lagoon. We then formed liposomes followed the course of protein production (the green fluorescent protein was used) inside and outside liposomes. Surprisingly, we found that some vesicles were able to synthesize the protein despite the low solute concentration, revealing that even a complex metabolic pathway spontaneously self-concentrates inside liposomes (Stano et al., submitted). These results provide a physical explanation to the origin of early functional cells, thanks to a favorable micro-environment created by liposomes, which can concentrate solutes inside their aqueous cavity.

### Vesicles assemblies

Current research on the origin of life typically focuses on the self-organization of molecular components in individual cell-like compartments, whereas no attempts have been made to investigate communities of compartments. Here we present our study on vesicles assemblies as a model of primitive cell communities.

At this aim, we firstly prepared giant vesicles (GVs) composed by a primitive lipid, namely oleic acid, which is negatively charged. GV membranes have been made stable by the addition of a neutral stabilizing phospholipid (POPC, 1-palmitoyl-2-oleoyl-*sn*-glycero-3-phosphatidylcholine). Oleate/POPC GV were prepared after the optimization of the “droplet transfer” method, which allows the facile entrapment of solutes like enzymes and DNA.

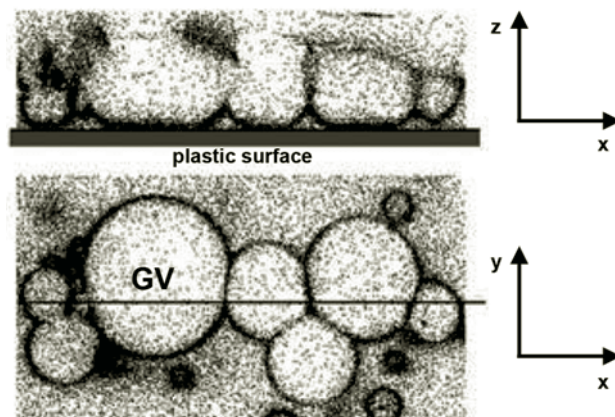
The addition of poly-L-arginine (PLA), a positively charged polypeptide, brings about the association of negatively charged oleate/POPC GV in form of small clusters first, that in turn grow to give rise to very large clusters. The formation of these GV “colonies” is stochastic but depends on the GV numerical density as well as on the PLA/GV ratio. Typically, these clusters contain up to 100 GV. GV are also destroyed after the addition of PLA (due to high local PLA concentration), and the maximal GV to clusters conversion is about 50%.

Once formed, we characterize the GV clusters properties. Phosphate bearing water-soluble solutes, such as ADP, fluorescent diphosphate and even t-RNA, present in the environment, readily penetrate into clusterized GV much faster than isolated GV. This is probably due to the help of PLA adsorbed between GV, which acts here like the well-known “cell penetrating peptides”. GV in the cluster can fuse with each other (fusion yield ca. 0.5-1%). Fusion was revealed by preparing clusters from two GV population, each containing a fluorescently labeled macromolecule. It is remarkable that vesicle fusion might provide a mechanism for

increasing the internal molecular complexity thanks to the acquisition of new molecules originally entrapped in another vesicle. Finally, GV clusters can capture new GV from the solution and grow. The new GV attach to the cluster border, demonstrating that PLA is still present in the outer GV cluster surface.

GV clusters are firmly attached to the solid support (an hydrophobic plastic polymer, in this case, and resist to hydraulic flow.

In summary, GV clusters display several interesting features, mimicking in very simple way bacterial colonies at the lowest possible complexity level. Intriguingly, bacterial L-forms (bacteria without cell walls) and GV display some common physico-mechanical behavior.



**Figure 2.** Sagittal ( $xz$ ) ( $130 \times 30 \mu\text{m}$ ) and horizontal ( $xy$ ) ( $130 \times 60 \mu\text{m}$ ) confocal images of GV associated to form “colonies”. The GV membranes have been stained by octadecyl rhodamine. Note the flattening of GV on the plastic surface. Reproduced from Carrara et al. (2012) with permission from Wiley.

### References

- Carrara, P., Stano, P., Luisi, P. L. (2012). Giant vesicles ‘colonies’: a model for primitive cell communities. *ChemBioChem*, 13:1497-1502.
- Luisi, P. L., Allegretti, M., Souza, T., Steiniger, F., Fahr, A., Stano, P. (2010). Spontaneous protein crowding in liposomes: A new vista for the origin of cellular metabolism. *ChemBioChem*, 11:1989-1992.
- Luisi, P. L., Ferri, F., Stano, P. (2006). Approaches to semi-synthetic minimal cells: a review. *Naturwissenschaften*, 93:1-13.
- Souza, T., Stano, P., Luisi, P. L. (2009). The minimal size of liposome-based model cells brings about a remarkably enhanced entrapment and protein synthesis. *ChemBioChem*, 10:1056-1063.
- Souza, T., Stano, P., Steiniger, F., D’Aguanno, E., Altamura, E., Fahr, A., Luisi, P. L. (2012). Encapsulation of ferritin, ribosomes, and ribopeptidic complexes inside liposomes: Insights into the origin of metabolism. *Orig. Life Evol. Biosph.*, 42:421-428.
- Souza, T., Steiniger, F., Stano, P., Fahr, A., Luisi, P. L. (2011). Spontaneous crowding of ribosomes and proteins inside vesicles: A possible mechanism for the origin of cell metabolism. *ChemBioChem*, 12:2325-2330.
- Stano, P., D’Aguanno, E., Bolz, J., Fahr, A., Luisi, P. L. (submitted). A remarkable self-organization process as the origin of primitive functional cells.

# Steady state analysis of a vesicle bioreactor with mechanosensitive channels

Ben Shirt-Ediss<sup>1,2</sup>, Ricard Solé<sup>1,4</sup> and Kepa Ruiz-Mirazo<sup>3,2</sup>

<sup>1</sup>ICREA-Complex Systems Lab, Institut de Biologia Evolutiva, CSIC-UPF, Barcelona, Spain

<sup>2</sup>Logic and Philosophy of Science Department, University of The Basque Country, Spain

<sup>3</sup>Biophysics Unit (CSIC-UPV/EHU), University of The Basque Country, Spain

<sup>4</sup>Santa Fe Institute, 1399 Hyde Park Road, Santa Fe NM 87501, USA

ben.shirt.ediss@gmail.com

## Abstract

An intermediate step between simple models of protocells having no membrane structure and the current, complex membrane organization of modern cells considers simple forms of membrane change that interact with the protocell dynamics in predictable ways. Here we analyse a kinetic model of a simple protocell system: a bioreactor vesicle involving mechanosensitive channels. This is a closed lipid bilayer which hosts a minimal metabolism  $X + E \rightarrow W + E$  inside its internal (variable volume) water pool, and with channels anchored in the membrane. The reactor can swell, opening the channels, which in turn allow enhanced passive diffusion of solutes  $X$  and  $W$  into and out of the reactor. We calculate under what external conditions and parameter regimes the reactor is able to maintain a far from equilibrium steady state, and what behaviours are possible in this reduced complexity scenario. This study is just a preliminary step in modelling the bigger question of how osmotic force sensing devices in early membranes could couple with early metabolisms, enhancing the stability of protocells in changing environments.

## Introduction

The study of protocellular structures, both from theoretical and experimental approximations, offers a unique opportunity to approach some of the key questions associated to the emergence of life in our biosphere (Deamer, 2011). Moreover, it also provides useful insight into different ways of approaching the problem of building synthetic compartments capable of reacting to external signals while carrying internal molecular structures (Solé et al., 2007). The potential repertoire of protocellular constructs is highly constrained by the type of membrane complexity that can be used. Membranes are the interface between the internal reactions and the external, usually fluctuating world. Part of the relevant phenomena that would take place depends on the ways membranes respond to environmental signals. Early models of protocells consider some sort of passive or active diffusion process with no or little complexity. However, it would be desirable to include within the protocell framework minimal models of membrane organization reacting to environmental cues in physically realistic ways. In this paper, we consider such a property by including mechanosensitive (MS) channels as a component of our model protocells.

MS channels, found in most free-living Bacteria and Archaea, are small functional protein nano-machineries which embed in lipid bilayers and open a water-filled pore in response to tension in the membrane (Kung, 2005; Sacklin, 1995; Kung et al., 2010). Modern day bacteria such as *Escherichia coli* are known to regulate their internal state with respect to a changing osmolarity environment by using various metabolic and transcriptional means, and the MS channels in their membranes are often cited as playing an “emergency release valve” role, quickly jettisoning solutes to avoid osmolysis when the environment rapidly dilutes (Booth and Blount, 2012). However, MS channels are thought to have evolved at a very early stage in the development of bacteria (Kloda and Martinac, 2002), and it is interesting to speculate what role osmotic force sensing devices in the membranes of protocells would have played in regulating their much simpler metabolisms in changing environments (Morris, 2002).

To make a departure point in this direction, this study analyses a minimal scenario where a vesicle acting as a miniaturised reactor – metabolising substrate  $X$  absorbed from the environment, into waste  $W$ , which is later expelled – can extend its viability over a wider range of external substrate concentrations, by incorporation of MS channels into the membrane. Although near-molecular detail dynamics models of MS channels in bilayers indeed exist (Louhivuoria et al., 2010, for example), our model is placed at the coarse grain level of chemical kinetics. This allows us to accommodate time scales of seconds as opposed to microseconds, and gives a coupled membrane-metabolism system amenable to some mathematical analysis. An overview of our modelling approach, where vesicles are modelled as a series of coupled reaction domains, has recently been given by Mavelli and Ruiz-Mirazo (2010).

## Bioreactor Kinetic Model

The bioreactor model (Figure 1) consists of a lipid vesicle floating in a reservoir. The reservoir is assumed so large that solute concentrations exterior to the vesicle are approximately constant. The internal aqueous pool of the vesicle



is modelled as a well-stirred chemical domain, and a semi-permeable lipid bilayer  $\lambda$  dm thick separates this domain from the reservoir. The surface areas ( $S_\mu$ ) of the water-facing sides of the inner and outer bilayer leaflets are assumed to be equal ( $\lambda \ll$  vesicle radius). The reservoir provides an inexhaustible source of substrate  $X$  at concentration  $[X]_{\text{out}}$  molar.

When the vesicle has no MS channels in the membrane, or when the channels are not yet open (Figure 1a),  $X$  diffuses passively (with diffusion constant  $D_X$ ) into the vesicle following Fick's Law. Physical space does not have a fully fledged representation in our model, and so this diffusion is taken to happen *only across* the bilayer region separating the domains. The vesicle membrane is assumed to be made of oleic acid and in what follows, the permeabilities of solutes  $X$  and  $W$  are set to be comparable to the permeability that oleate vesicles exhibit to ribose (Sacerdote and Szostak, 2005), which defines a realistic time scale in seconds.

Once inside the internal aqueous pool of the vesicle,  $X$  is enzymatically transformed (e.g. isomerized) into product  $W$  at rate  $k$ . To keep complexity to the minimum possible, the responsible enzyme  $E$  is not explicitly modelled, but is considered trapped inside the vesicle pool facilitating the  $X \rightarrow W$  reaction, which by contrast does not take place in the external reservoir. The reverse rate of the reaction is considered negligible and disregarded.

After  $W$  is produced, it passively diffuses out of the vesicle, but with a different diffusion constant  $D_W$ . The environment is a sink for  $W$  and so at all times  $[W]_{\text{out}} = 0$ . Together with the above assumptions, this guarantees that the mass flow through the reactor is always in one direction (left to right, Figure 1) or zero.

The bioreactor model is thus a two variable dynamical system characterised by the number of  $X$  and  $W$  species inside the vesicle aqueous pool. With some terms to be explained shortly, the state vector evolves from initial condition  $\langle X_{\text{in}}^0, W_{\text{in}}^0 \rangle$  according to the following coupled ODEs:

$$\frac{dX_{\text{in}}}{dt} = (D_X S_\mu + \sigma(T_\mu) D_c N_c) \frac{[X]_{\text{out}} - [X]_{\text{in}}}{\lambda} - k X_{\text{in}} \quad (1)$$

$$\frac{dW_{\text{in}}}{dt} = k X_{\text{in}} - (D_W S_\mu + \sigma(T_\mu) D_c N_c) \frac{[W]_{\text{in}}}{\lambda} \quad (2)$$

where  $[X]_{\text{in}}$  and  $[W]_{\text{in}}$  are the molar concentrations of  $X$  and  $W$  in the vesicle pool respectively, written as  $[X]_{\text{in}} = X_{\text{in}}/\mathcal{V}$  and  $[W]_{\text{in}} = W_{\text{in}}/\mathcal{V}$ . Here term  $\mathcal{V} = N_A V$  is called the *normalised volume* of the vesicle aqueous pool, and is used in the analysis section in preference to the actual litre volume  $V$ .

It is important to note that the volume of the vesicle aqueous pool is *not constant*, but changes in time as water moves across the vesicle membrane through osmosis. Owing to its

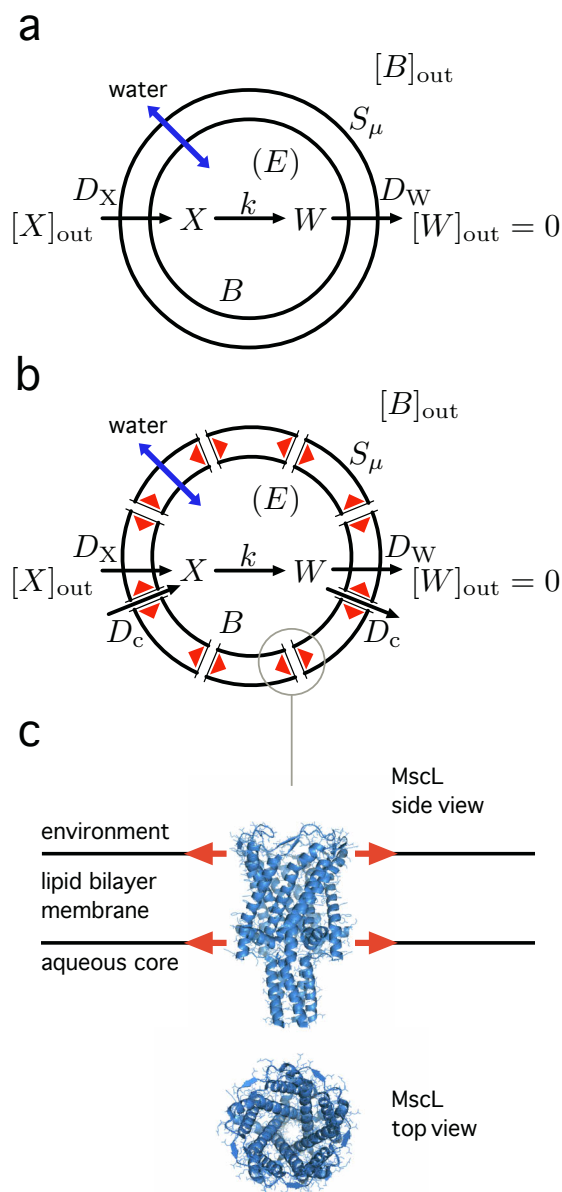


Figure 1: Bioreactor schematic. (a) A vesicle with no MS channels, or with MS channels but insufficient membrane tension to open them and (b) an osmotically stressed vesicle with MS channels open, permitting enhanced passive transport rates for solutes  $X$  and  $W$  into/out of the vesicle (red arrows denote membrane tension). (c) Molecular structure of a contemporary MS channel of type MscL (mechanosensitive channel of large unitary conductance), typically found in *E. coli* membranes, pictured here in closed conformation.

very high permeability, water is assumed to flow across the membrane instantaneously and the vesicle volume is set at every moment such that the total concentration of molecular species inside the vesicle is equal to the total concentration

outside:

$$[B]_{\text{in}} + [X]_{\text{in}} + [W]_{\text{in}} = [B]_{\text{out}} + [X]_{\text{out}} \quad (3)$$

which gives the normalised volume of the vesicle at each instant as:

$$\mathcal{V} = \frac{B_{\text{in}} + X_{\text{in}} + W_{\text{in}}}{[B]_{\text{out}} + [X]_{\text{out}}} \quad (4)$$

A fixed number of buffer molecules  $B$  are trapped inside the vesicle aqueous pool for stability purposes. These are non-reactive molecules which cannot permeate the lipid bilayer, and likewise they exist in the environment at constant concentration  $[B]_{\text{out}}$ . In this work, we choose to set the number of  $B$  in the internal aqueous pool such that the vesicle is perfectly spherical when the outside substrate concentration  $[X]_{\text{out}} = 0$  and there are no solutes  $X$  or  $W$  inside the reactor. That is,  $B_{\text{in}}$  is set such that the initial volume  $\mathcal{V}_0 = B_{\text{in}}/[B]_{\text{out}}$  gives  $\Phi = 1$  (see below).

The extent to which the vesicle is swelled, or deflated, is captured by a non-dimensional measure called the *reduced surface*:

$$\Phi = \frac{S_{\mu}}{\sqrt[3]{36\pi(\mathcal{V}/N_A)^2}} \quad (5)$$

defined as the ratio of the actual surface area of the vesicle, to the surface area corresponding to the volume of the internal aqueous pool when that volume is considered spherical in shape. When  $\Phi > 1$  the system is deflated with an undefined shape and has surplus surface area for the volume. When  $\Phi = 1$ , the vesicle is perfectly spherical, and when  $\Phi < 1$ , the vesicle is spherical but in a state of membrane tension because insufficient surface exists to perfectly wrap the volume. The reduced surface allows us to define the percentage membrane tension  $0 \leq T_{\mu} \leq 1$  as follows:

$$T_{\mu} = \begin{cases} \frac{1-\Phi}{\epsilon}, & \text{if } 1 - \epsilon \leq \Phi \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where the lipid membrane ruptures from excess osmotic pressure and the vesicle no longer exists if  $\Phi < 1 - \epsilon$ . This non-dimensional parameter  $\epsilon$  (the burst tolerance) therefore establishes the stability boundary of the system.

When sufficient, the membrane tension  $T_{\mu}$  activates any existing MS channels in the membrane (Figure 1b), enhancing the passive diffusion rates of  $X$  and  $W$  through the membrane. Unlike normal passive diffusion through the bilayer, where diffusion rate is influenced by specific molecular properties of solutes, the channels open water-filled pores and so are assumed to augment the passive diffusion rates of  $X$  and  $W$  equally. The  $N_c$  membrane channels are each modelled to open continuously at a percentage which is a linear function of membrane tension:

$$\sigma(T_{\mu}) = \begin{cases} 0, & \text{if } T_{\mu} \leq T_0 \\ 1, & \text{if } T_{\mu} \geq T_m \\ \frac{T_{\mu}-T_0}{T_m-T_0} & \text{otherwise} \end{cases} \quad (7)$$

for  $T_0 < T_m$ , where  $T_0$  and  $T_m$  are the percentages of membrane tension when the channels begin to open, and are maximally open, respectively. If  $T_m$  is only marginally larger than  $T_0$ , then the channels will snap open more or less instantaneously as  $T_0$  is reached. When tension in the membrane exceeds  $T_m$ , the channels are maximally extended and do not open further with increasing tension. In reality, MS channels don't open linearly, but rather open in a quantum manner through a series of mid-states. Furthermore, channels can 'flicker' back to these mid-states once opened (Sukharev et al., 2001). Considering these facts, our measure  $\sigma(T_{\mu})$  can be interpreted as the average percent conductivity per channel in the membrane at tension  $T_{\mu}$ .

When the  $N_c$  channels are fully open, the membrane diffusion constant for a solute, for example  $X$ , increases by  $1 + (D_c N_c / D_X S_{\mu})$  fold, where  $D_c$  is the effective diffusion constant for a single open channel.

Importantly, the MS channels are assumed not to permit the passage of buffer  $B$  or enzyme  $E$  into/out of the vesicle compartment. Finally, lipid flux to and from the lipid bilayer membrane is not explicitly modelled, but it is assumed that the bilayer exists in a solution where the outside lipid concentration is close to the CVC for oleic acid and thus the inside/outside vesicle surface area remains approximately stable over time at  $S_{\mu}$ . Thus, there is no possibility for membrane growth in this simplified model; the surface area is assumed not to increase under osmotic tension conditions. The parameters used in the model can be found in Table 1 below and in the figure captions.

| Parameter           | Value                       | Unit                                         |
|---------------------|-----------------------------|----------------------------------------------|
| $S_{\mu}$           | $3.1416 \times 10^{-12}$    | $\text{dm}^2$                                |
| $\lambda$           | $4.0 \times 10^{-8}$        | $\text{dm}$                                  |
| $\epsilon$          | 0.5                         |                                              |
| $D_{\text{ribose}}$ | $2.65 \times 10^8$          | $\text{dm}^2 \text{s}^{-1} \text{mole}^{-1}$ |
| $D_c$               | $D_{\text{ribose}} S_{\mu}$ | $\text{dm}^4 \text{s}^{-1} \text{mole}^{-1}$ |
| $N_c$               | 3                           |                                              |
| $k$                 | 0.009                       | $\text{s}^{-1}$                              |
| $[B]_{\text{out}}$  | 0.2                         | $\text{mole dm}^{-3}$                        |
| $B_{\text{in}}$     | 63064                       |                                              |

Table 1: Model parameters.  $S_{\mu}$  is set to the external surface area of a 50nm radius sphere, the typical surface area of an extruded unilamellar oleic acid vesicle.  $D_{\text{ribose}}$  is from (Mavelli and Ruiz-Mirazo, 2010), derived from experimental data in (Sacerdote and Szostak, 2005). Constants  $D_c$  and  $N_c$  are set such that when all MS channels are open, the membrane permeability to ribose increases 4-fold.

## Steady State Analysis

It is instructive to analyse the steady state system behaviour in two parts. The first part applies when the vesicle has *constant membrane permeability* with respect to membrane tension, namely (a) when the reactor has no MS channels, (b) when the reactor has channels but insufficient membrane tension to open them (in both cases,  $\sigma(T_\mu) = 0$ ) and also (c) when the bioreactor has maximally open channels that are no longer responsive to changes in membrane tension ( $\sigma(T_\mu) = 1$ ). The second part completes the picture by finding the steady states when the MS channels are partially open ( $0 < \sigma(T_\mu) < 1$ ) and membrane permeability depends on membrane tension.

### Reactor with constant membrane permeability

The bioreactor is defined as being in steady state when (i) the number  $X$  and  $W$  species inside are not changing (metabolic steady state), whilst at the same time (ii) osmotic equilibrium (3) exists.

Given a fixed outside substrate concentration  $[X]_{\text{out}} \geq 0$ , our objective will be to solve the model for the steady state vesicle volume  $\mathcal{V}^*$  which ensures both (i) and (ii). Normalised volume  $\mathcal{V} = N_A V$  is useful to use, since it both represents the vesicle litre volume  $V$  and provides a convenient way to convert internal species numbers to molar concentrations (by dividing by  $\mathcal{V}$ ) and vice versa. Once  $\mathcal{V}^*$  is known, all other properties of the system at steady state can be derived.

Dealing with condition (i) first, for a bioreactor where  $\sigma(T_\mu) = 0$  (no channels, or insufficient tension to open existing channels), setting (1) and (2) to 0 yields the steady state concentrations of  $X$  and  $W$  inside the vesicle, given that the vesicle volume is  $\mathcal{V}$  (a value which we are free to choose arbitrarily at this stage):

$$[X]_{\text{in}}^* = \frac{[X]_{\text{out}}}{1 + \mathcal{V}\alpha} \quad (8)$$

$$[W]_{\text{in}}^* = \frac{\beta\mathcal{V}[X]_{\text{out}}}{1 + \mathcal{V}\alpha} \quad (9)$$

where constants  $\alpha = \frac{k\lambda}{D_X S_\mu}$  and  $\beta = \frac{k\lambda}{D_W S_\mu}$ . When these internal concentrations are present,  $X$  is entering the vesicle at the same rate that it is being transformed to  $W$ , which in turn is equal to the rate that  $W$  is leaving the vesicle. For a bioreactor where  $\sigma(T_\mu) = 1$  (maximally open channels), the solutions are the same, but with constant  $\alpha$  replaced by constant  $\gamma = \frac{k\lambda}{D_X S_\mu + D_c N_c}$ , constant  $\beta$  replaced by  $\eta = \frac{k\lambda}{D_W S_\mu + D_c N_c}$ , and the requirement that volume  $\mathcal{V}$  is large enough to make a membrane tension  $T_\mu \geq T_m$ , i.e. sufficient to fully open the channels.

Now, we further constrain  $\mathcal{V}$  so that the osmotic equilibrium requirement (ii) is additionally satisfied. Substituting (8) and (9) into (3) gives:

$$\frac{B_{\text{in}}}{\mathcal{V}} + \frac{[X]_{\text{out}}}{1 + \mathcal{V}\alpha} + \frac{\beta\mathcal{V}[X]_{\text{out}}}{1 + \mathcal{V}\alpha} = [B]_{\text{out}} + [X]_{\text{out}} \quad (10)$$

which when re-arranged leads to a quadratic equation in  $\mathcal{V}$ , the solution to which yields  $\mathcal{V}^*$ , given that the outside concentration is  $[X]_{\text{out}}$ :

$$\mathcal{V}^* = \frac{(\alpha B_{\text{in}} - [B]_{\text{out}}) \pm \sqrt{\Delta}}{2a} \quad (11)$$

where coefficient

$$a = \alpha[B]_{\text{out}} + (\alpha - \beta)[X]_{\text{out}} \quad (12)$$

and the discriminant

$$\Delta = ([B]_{\text{out}} - \alpha B_{\text{in}})^2 + 4B_{\text{in}}\alpha[B]_{\text{out}} + 4B_{\text{in}}(\alpha - \beta)[X]_{\text{out}} \quad (13)$$

which becomes negative when

$$([B]_{\text{out}} - \alpha B_{\text{in}})^2 + 4B_{\text{in}}\alpha[B]_{\text{out}} < 4B_{\text{in}}(\beta - \alpha)[X]_{\text{out}} \quad (14)$$

Substituting  $\mathcal{V}^*$  into (8) and (9) reconstitutes the full state vector of the system at steady state, and substituting it into (5) gives the corresponding  $\Phi^*$ , used in the figures.

The steady state behaviour of the bioreactor is dependent on the magnitude of the diffusion constants  $D_X$  and  $D_W$  with respect to each other (Figure 2). When  $X$  and  $W$  passively diffuse at equal rates ( $D_X = D_W$ ) the reactor will always be perfectly spherical at steady state, independent of the outside substrate concentration  $[X]_{\text{out}}$ . This is because when  $\alpha = \beta$  then (12) becomes independent of  $[X]_{\text{out}}$  and (11) reduces to  $\mathcal{V}^* = B_{\text{in}}/[B]_{\text{out}} = \mathcal{V}_0$  for the parameters used. In this state  $\Phi = 1$ , thus there is no membrane tension to activate MS channels.

Conversely, when  $D_W > D_X$ , the reactor actually shrinks in volume as  $[X]_{\text{out}}$  is increased (Figure 2, inset). Thus, whenever waste  $W$  is more permeable than substrate  $X$ , the membrane will never have tension at steady state.

The most interesting case is when  $D_X > D_W$ . Here, increasing  $[X]_{\text{out}}$  swells the volume of the bioreactor, causing tension in the membrane. Figure 2 (solid black line) shows that the bioreactor follows a succession of steady states at decreasing  $\Phi$  values (increasing membrane tension) as  $[X]_{\text{out}}$  is increased. For the parameter values used, a catastrophic value of  $[X]_{\text{out}} = [X]_{\text{out}}^c$  also exists (vertical red line), passed which the system has no viable steady states, and will burst. This point corresponds to the discriminant (13) becoming negative and complex roots for  $\mathcal{V}^*$ . This cannot happen in the  $D_W > D_X$  regime, because when  $\beta < \alpha$ , equality (14) can never be true (the left hand side is always positive).

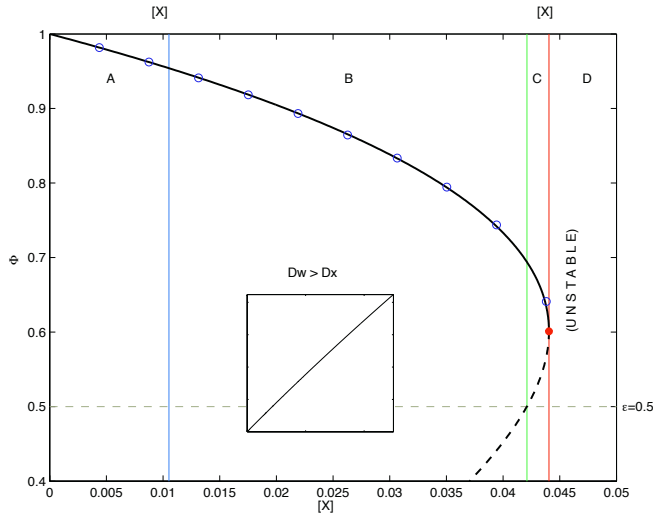


Figure 2: Steady state  $\Phi^*$  for a bioreactor vesicle with no channels, as  $[X]_{\text{out}}$  increases. Main figure:  $D_X = 2.0 \times D_{\text{ribose}} > D_W = 0.1 \times D_{\text{ribose}}$ . In region A, the bioreactor has one stable node, in regions B and C it has an additional unstable saddle point (but in region B, this point is outside the viability zone) and in region D the reactor has no steady state. Inset: steady state behaviour when magnitudes of diffusion constants reversed ( $D_X < D_W$ ) involves the reactor shrinking rather than swelling. Parameters given in Table 1. Blue circles validate the analysis by showing mean equilibrated values of  $\Phi$  from stochastic simulations of the model with the Gillespie algorithm (subsequently performed as validation in all plots).

Thus, when  $D_X > D_W$  the system also has the possibility for an *intrinsic limit* to the maximum outside substrate concentration, at which steady state is possible. This means that if  $[X]_{\text{out}}$  is gradually increased from 0, the transition from zero membrane tension to maximum membrane tension and burst (when  $\Phi < 1 - \epsilon$ ) is not necessarily smooth: the reactor can become prematurely unstable if  $[X]_{\text{out}}^C$  is encountered along the way (illustrated by the red dot in Figure 2) before reaching the viability threshold, determined by  $\epsilon = 0.5$ .

For an arbitrary set of reactor parameters  $\{D_X, k, D_W\}$ , the normalised reactor volume for which a catastrophic burst can be achieved,  $\mathcal{V}_{\text{min}}^C$ , can be shown to have a lower limit. It is derived by setting the discriminant to 0 in (11), which makes  $\mathcal{V}^* = \mathcal{V}^C$ , and then taking the limit of the resulting expression as  $\alpha$  tends to zero, which corresponds to diffusion constant  $D_X$  tending to infinity, or reaction rate  $k$  tending to 0:

$$\mathcal{V}_{\text{min}}^C = \lim_{\alpha \rightarrow 0} \mathcal{V}^C = \frac{2B_{\text{in}}}{[B]_{\text{out}}} \quad (15)$$

which can be written  $\mathcal{V}_{\text{min}}^C = 2\mathcal{V}_0$  for our choice of pa-

rameters, corresponding to  $\Phi = 0.63$ . We can guarantee that catastrophic burst cannot happen for membrane tensions less than this.

Another interesting occurrence when  $D_X > D_W$  is the appearance of an unstable saddle point<sup>1</sup> (Figure 2, dotted line). This is not always present, but happens when (12) changes sign with increasing  $[X]_{\text{out}}$ , to become negative after an initially positive value at  $[X]_{\text{out}} = 0$ . Re-arranging (12), we see that an additional unstable saddle point accompanying the existing stable node appears when:

$$[X]_{\text{out}} > \frac{\alpha}{\beta - \alpha} [B]_{\text{out}} \quad (16)$$

This point is marked on the figures as  $[X]_{\text{out}}^S$  (vertical blue lines), for the set of parameter values used. The presence of an unstable point means that not all initial conditions  $\langle X_{\text{in}}^0, W_{\text{in}}^0 \rangle$  will lead to the stable node of the system. For membrane burst tolerance  $\epsilon = 0.5$ , the unstable saddle point only becomes reachable at higher values of  $[X]_{\text{out}}$  (green vertical lines on figures).

### Reactor with MS channels

The full bioreactor model with MS channels in the membrane is actually a hybrid dynamical system. When the channels are closed, or maximally open, the system behaves as a reactor with constant membrane permeability as discussed above. In the tension region  $T_0 < T_\mu < T_m$  where the channels are opening in response to membrane tension (and thus membrane permeability is *dependent* on membrane tension), the evolution equations of the system change in the sense that function  $\sigma(T_\mu)$  is now not constant. This makes the calculation of metabolic steady state (i) in this region more cumbersome:

$$[X]_{\text{in}}^* = \frac{[X]_{\text{out}}(k_1 - k_3\mathcal{V})}{(k_4 - k_3)\mathcal{V} + k_1} \quad (17)$$

$$[W]_{\text{in}}^* = \frac{[X]_{\text{out}}\mathcal{V}(k_1k_4 - k_3k_4\mathcal{V})}{(k_3k_3 - k_3k_4)\mathcal{V}^2 + (k_4k_2 - k_3k_2 - k_3k_1)\mathcal{V} + k_1k_2} \quad (18)$$

where constants  $k_1, \dots, k_4$  are defined in the Appendix. Function  $\sigma(T_\mu)$  contains the non-linear term  $\Phi$  (5) from which  $\mathcal{V}$  cannot easily be isolated, and therefore the above result entails making a linear approximation of  $\Phi$  in terms of  $\mathcal{V}$ , also detailed in the Appendix.

When equations (17) and (18) are substituted into water balance equation (3) to find  $\mathcal{V}^*$  at which osmotic equilibrium (ii) is also present, a quartic equation in  $\mathcal{V}$  results (as opposed to a quadratic, before). This quartic equation is not stated, but the important point is that it has one viable solution which connects the steady state solutions for

<sup>1</sup>These unstable points have a Jacobian matrix with one zero and one negative eigenvalue.



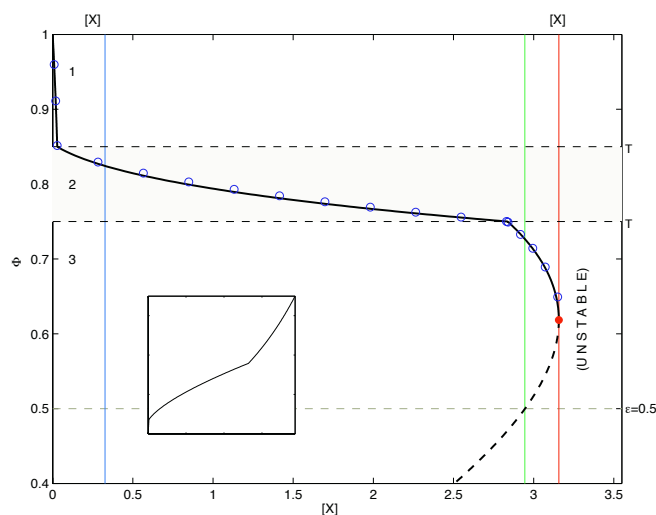


Figure 3: Steady state  $\Phi^*$  for a bioreactor vesicle with  $N_c = 3$  MS channels, as  $[X]_{\text{out}}$  increases. Main figure: The presence of channels extend the  $[X]_{\text{out}}$  viability range of the reactor by around 70 fold, as compared to Figure 2. In region 1, channels remain closed, in region 2 (shaded) channels open in response to membrane tension and in region 3, channels are maximally open. Inset: Combinations of steady state  $X$  and  $W$  molecule numbers in the aqueous pool of the reactor, during the channels closed, channels opening, and channels maximally open phases. Parameters are identical to Figure 2, with MS channels active in the range  $0.85 > \Phi > 0.75$ .

$\sigma(T_\mu) = 0$  to the solutions for  $\sigma(T_\mu) = 1$  over the tension region  $T_0 < T_\mu < T_m$ , as illustrated in Figure 3, between the horizontal dotted lines. After tension  $T_m$  is exceeded, the system has again a fixed membrane permeability, and continuing to further increase  $[X]_{\text{out}}$  causes the catastrophic limit  $[X]_{\text{out}}^c$  to be passed (Figure 3, red vertical line), where the steady state of the system disappears. Figure 4 shows what actually happens to the dynamics of the three flows through the reactor, once  $[X]_{\text{out}}^c$  is passed.

If the channels of the reactor remain closed until a higher membrane tension, closer to the critical limit of  $\Phi = 1 - \epsilon = 0.5$ , then for the parameters used, the reactor suffers a negative discriminant (13) *before* the channels become active. This interesting situation is shown in Figure 5 and results in a small window of  $[X]_{\text{out}}$  values for which the reactor has two stable states. One stable state involves the channels being closed and the reactor remaining in a ‘low throughput’ mode. The other involves the channels being slightly open and the reactor in a ‘high throughput’ mode. It can also be seen that the reactor will exhibit hysteresis in switching between the two steady states as  $[X]_{\text{out}}$  is increased and decreased across this region.

To conclude the study, Figure 6 compares the viability region of a bioreactor with no channels to a bioreactor with MS channels, when  $[X]_{\text{out}} = 1.0\text{M}$ , for a large parameter space of diffusion constants  $D_X$ ,  $D_W$  and reaction rates  $k$ . Here, ‘viability’ means the capacity to hold a far from equilibrium steady state at the imposed external substrate concentration. It can be seen that inclusion of MS channels improves the viability range of the bioreactor for *intermediate* values of reaction rate  $k$  (middle left and middle right plots). Nevertheless, when  $k$  is high and the internal reaction fast (top left plot), the viability region  $D_X > D_W$  is very small indeed and the presence of channels (orange region) does not extend this much, meaning that the viability is similarly limited. Conversely, when  $k$  is low (bottom right plot), then MS channels are generally not required to maintain a steady state for the range of  $\{D_X, D_W\}$  combinations. Thus again, the two reactors have equal viability.

## Discussion and Conclusions

In this study, we aimed to model the simplest protocellular scenario where the action of MS channels could be explored. We found that our minimal system can exhibit at least two

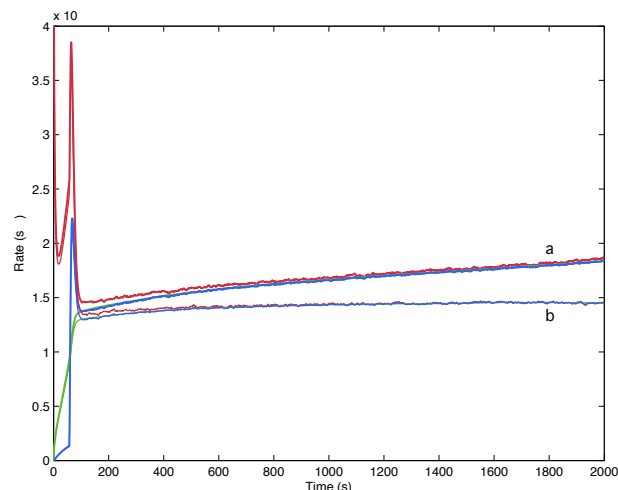


Figure 4: Reactor dynamics when passing the catastrophic threshold. The rates of the three flows through the reactor in Figure 3 are shown, when (a)  $[X]_{\text{out}} = 3.2275\text{M}$  exceeds and (b) when  $[X]_{\text{out}} = 3.0725\text{M}$  does not exceed the catastrophic external substrate concentration  $[X]_{\text{out}}^c = 3.1569\text{M}$ . In both cases, red lines are the rate at which  $X$  enters the vesicle, green lines the rate that  $X$  transforms into  $W$  and blue lines the rate at which  $W$  leaves the vesicle. In (b) all three rates manage to become equilibrated in contrast to (a) where a small difference persists between them, leading to the accumulation of  $X$  and  $W$  inside the vesicle and eventual osmotic rupture. The stochastic simulation was performed with the Gillespie algorithm.

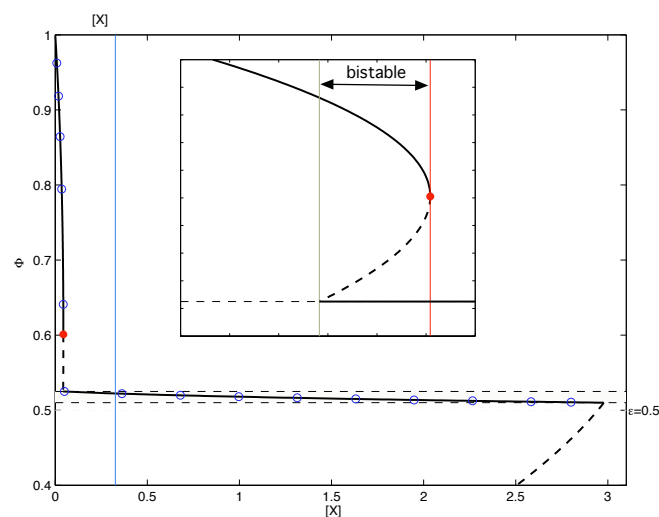


Figure 5: Bi-stability in the bioreactor with MS channels. Main figure: Steady states of the same bioreactor as in Figure 3, but now with channels activating at higher membrane tension  $0.525 > \Phi > 0.51$ . Inset: a zoomed view of the small bistable region on the left of the figure, showing two stable nodes exist in the range  $0.0429M < [X]_{\text{out}} < 0.0440M$ : one node is in the channels closed regime and the other is in the channels active regime.

interesting behaviours which were not obvious from the outset. The first is that the reactor can suddenly burst when the external substrate concentration reaches a critical concentration, even though it may be far from maximum membrane tension at that point. Secondly, we found that under some parameter regimes, the bioreactor with channels can have two viable steady states for the same external substrate concentration. In this scenario, perturbations applied inside the reactor could flip it like a toggle switch between a low throughput mode and a higher throughput mode.

Importantly, we observe that the bioreactor demonstrates the above behaviours only when substrate  $X$  is more permeable than product  $W$ , otherwise the reactor will remain spherical or shrink with increasing external substrate concentration<sup>2</sup>.

Our general hypothesis that MS channels would improve the stability region of the reactor was confirmed, but only partially. We found that high or low values of the reaction rate constant  $k$  bring about the same viability region, whether or not the reactor included channels. Only for intermediate values of  $k$  did having channels significantly widen the range of external substrate concentrations the reactor

<sup>2</sup>It must be noted when  $D_W \geq D_X$ , then for high  $[X]_{\text{out}}$ , even though the reactor is stable in principle, unrealistic molecule numbers could exist inside the vesicle volume, invalidating the well-stirred assumption in this domain.

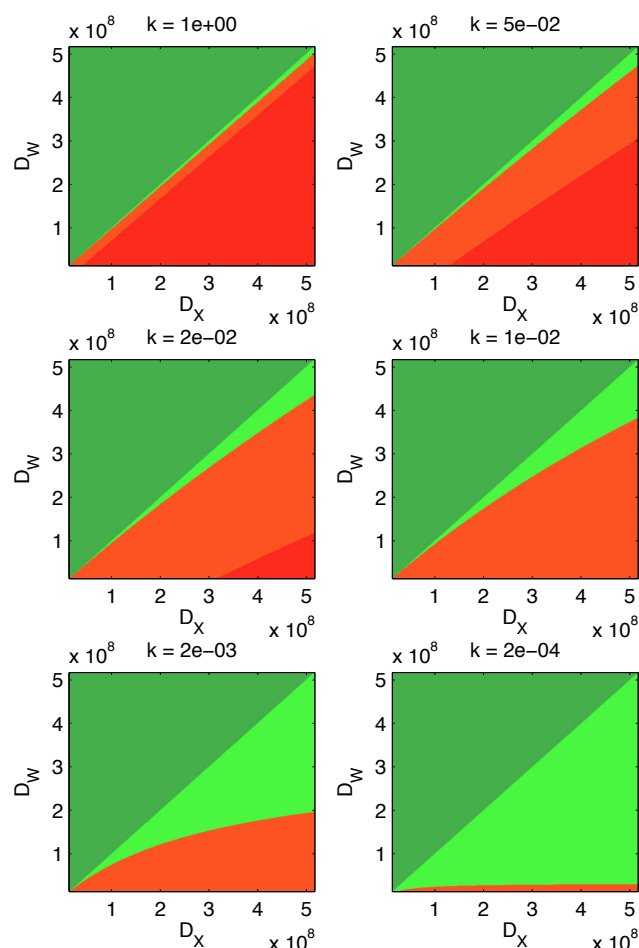


Figure 6: Comparison of the viability of a bioreactor with no channels to a bioreactor with MS channels, over a region of parameter space. Each plot is for a different value of reaction rate constant  $k$ . For outside substrate concentration  $[X]_{\text{out}} = 1.0M$ : (a) dark green and light green areas indicate that both reactors have a steady state at the particular  $\{D_X, k, D_W\}$  parameter combination, with the dark green area indicating this stability is simply because the reactor does not actually go into a tension state ( $D_W \geq D_X$ ), (b) orange areas indicate where *only* the MS channels reactor has a steady state when  $[X]_{\text{out}} = 1.0M$ , finally (c) red areas show parameter combinations where neither reactor can achieve steady state for the imposed external concentration. Diffusion constants explored are  $\pm 95\%$  that of  $D_{\text{ribose}}$ . Model parameters are different from previous figures in that burst tolerance  $\epsilon = 0.1$  and in the reactor with channels, MS are active between  $0.92 > \Phi > 0.91$  ( $T_o = 0.8, T_m = 0.9$ ).

could handle.

We believe that our results might be of relevance to future work involving this type of membrane organization,

perhaps within the context of simple vesicles (such as liposomes) carrying transmembrane properties with MS properties. Owing to their simple but non-linear responses, MS open new avenues within the study of synthetic biological systems (Cheng and Lu, 2012; Maharbiz, 2012). Since MS channels can help to modulate the protocell response to signals, allowing them to achieve more complex behaviors (such as alternative steady states), and MS can also be modulated, for example, by means of light activation (Folgering et al., 2004), we conjecture that MS channels can play a role in implementing computational features in simple models or liposome computation (Smaldon et al., 2010).

## Appendix

In general, a linear approximation of  $\Phi$  (5) is not possible. However, for the small range of  $\mathcal{V}$  values that the channels are active, this approximation is reasonable. We can write  $\Phi = m\mathcal{V} + c$  where

$$m = \frac{\Phi_o - \Phi_m}{\mathcal{V}_o - \mathcal{V}_m} \quad (19)$$

$$c = \Phi_o - m\mathcal{V}_o \quad (20)$$

Constant  $\Phi_o$  is obtained by inverting (6) and setting  $T_\mu = T_o$ . In turn, constant  $\mathcal{V}_o$  is obtained by inverting (5) and setting  $\Phi = \Phi_o$ . Constants  $\Phi_m$  and  $\mathcal{V}_m$  are calculated in similar ways.

The constants used in (17) and (18) are:

$$k_1 = D_X S_\mu k_5 + D_c N_c k_6$$

$$k_2 = D_W S_\mu k_5 + D_c N_c k_6$$

$$k_3 = D_c N_c m$$

$$k_4 = k\lambda k_5$$

$$k_5 = \epsilon(T_m - T_o)$$

$$k_6 = 1 - T_o\epsilon - c$$

## References

- Booth, I. R. and Blount, P. (2012). The mscs and mscl families of mechanosensitive channels act as microbial emergency release valves. *Journal of Bacteriology*, 194(18):4802–4807.
- Cheng, A. and Lu, T. K. (2012). Synthetic biology: An emerging engineering discipline. *Annu. Rev. Biomed. Eng.*, 14:155–178.
- Deamer, D. (2011). *First Life: Discovering the Connections Between Stars, Cells, and How Life Began*. University of California Press.
- Folgering, J. H. A., Kuiper, J. M., de Vries, A. H., Engberts, J. B. F. N., and Poolman, B. (2004). Lipid-mediated light activation of a mechanosensitive channel of large conductance. *Langmuir*, 20:6985–6987.
- Kloda, A. and Martinac, B. (2002). Common evolutionary origins of mechanosensitive ion channels in archaea, bacteria and cell-walled eukarya. *Archaea*, 1:35–44.
- Kung, C. (2005). A possible unifying principle for mechanosensation. *Nature*, 436:647–654.
- Kung, C., Martinac, B., and Sukharev, S. (2010). Mechanosensitive channels in microbes. *Annu. Rev. Microbiol.*, 64:313–329.
- Louhivuoria, M., Risselada, H. J., van der Giessen, E., and Marrink, S. J. (2010). Release of content through mechano-sensitive gates in pressurized liposomes. *PNAS*, 107(46):19856–19860.
- Maharbiz, M. M. (2012). Synthetic multicellularity. *Trends in Cell Biology*, 22(12):617–623.
- Mavelli, F. and Ruiz-Mirazo, K. (2010). Environment: a computational platform to stochastically simulate reacting and self-reproducing lipid compartments. *Phys. Biol.*, 7:1–13.
- Morris, C. E. (2002). How did cells get their size? *Anat. Rec.*, 268:239–251.
- Sacerdote, M. G. and Szostak, J. W. (2005). Semipermeable lipid bilayers exhibit diastereoselectivity favoring ribose. *PNAS*, 102(17):6004–6008.
- Sacklin, H. (1995). Mechanosensitive channels. *Annu. Rev. Physiol.*, 57:333–353.
- Smaldon, J., Romero-Campero, F. J., Trillo, F. F., Gheorghe, M., Alexander, C., and Krasnogor, N. (2010). A computational study of liposome logic: towards cellular computing from the bottom up. *Syst. Synth. Biol.*, 4:157–179.
- Solé, R. V., Munteanu, A., Rodriguez-Caso, C., and Macía, J. (2007). Synthetic protocell biology: from reproduction to computation. *Phil. Trans. R. Soc. B*, 362:1727–1739.
- Sukharev, S., Betanzos, M., Chiang, C.-S., and Guy, H. R. (2001). The gating mechanism of the large mechanosensitive channel mscl. *Nature*, 409:720–724.

# How Fast Can We Evolve Something?

Inman Harvey

Evolutionary and Adaptive Systems Group  
University of Sussex, Brighton, UK  
inmanh@gmail.com

## Abstract

When applying a Genetic Algorithm to a new problem, how many generations should one reasonably expect to wait before reaching an acceptable solution? Using arguments based on information flow into the genepool, mediated by selection each generation, we present a rough and ready heuristic.

## The Problem

You have encoded a new problem for solution by Genetic Algorithm (GA) in what you hope is a sensible manner, using a population size and mutation rates that appear appropriate. Yet after 10,000 generations you are nowhere near reaching an acceptable solution; was that long enough to wait?

The textbooks and standard literature provide surprisingly little advice on this crucial matter. Using arguments inspired by information-theoretic approaches to evolution (Kimura, 1961; Worden, 1995), we present a crude Rule of Thumb. Primarily we are concerned with complex problems associated with a fixed high-dimensional fitness landscape, but we start by recasting the problem in terms of a childrens' game.

## The Game of Twenty Questions

For convenience, suppose there is a pre-agreed list of  $2^{20}$  objects from which Player 1 secretly picks one as the target. Player 2 may ask 20 binary questions ('animal or not?', 'bigger than a brick or not?') to try and identify the target; what is the best strategy? Ideally, each successive question should divide the currently remaining pool of possibilities exactly in half, so that Player 1's response reduces the remaining uncertainty of Player 2 by the maximum possible 1 bit of Shannon information. Any question that fails to divide the pool into halves will provide less information and may leave some remaining uncertainty after the 20<sup>th</sup> question.

This can be reformulated into a simplistic asexual GA. With a population size of  $2^{20}$ , the initial generation consists of one example of every possible target (that could be represented by different binary genotypes length 20). Each question corresponds to a generation with a round of truncation selection where the 'fitter' half of the population is selected, and then duplicated (without mutation) to maintain the same population size. After 20 such generations, if the selective 'questions' have been ideally posed, the final population should consist of  $2^{20}$  copies of the desired target.

Clearly the very maximum amount of information, passed through such 50% truncation selection into the population (or its genepool), is 20 bits in 20 generations.

A more realistic asexual GA will achieve significantly less than this 1 bit information flow per generation. Selection is unlikely to produce the ideal 50/50 split; also the population will typically be far smaller than the complete genotype space. Thus mutation must partially compensate by expanding the population each generation beyond the limited set of possibilities within the initial small population; this is at the expense of some information being lost from the genepool each generation. Hence, as Worden (1995) points out, the figure of 1 bit per generation counts as a speed limit, like the speed of light, that is an idealized upper bound; in practice, significantly fewer bits per generation are achievable.

Higher rates of selection such as 25% truncation selection would result in 2 bits per generation as this upper speed limit. But since there are other reasons (such as avoiding getting trapped on local optima) that tend to make higher rates undesirable, we focus here on the paradigm case of 50% truncation selection. This is equivalent to the most basic forms of tournament selection (Harvey, 2011).

**Redundancy.** The example of binary genotypes of length 20 does indeed imply a search-space of size  $2^{20}$ . However in evolutionary search we are often not searching for just one target genotype, but any one of an equivalence class that count as equally fit. In the game of 20 Questions, if the  $2^{20}$  possible targets actually fell into say  $2^{10}$  different equivalence classes, and the second Player can win by identifying the equivalence class rather than the unique target, then the initial uncertainty can be counted as 10 bits rather than 20 bits. This clearly reduces the number of Questions (or generations) needed.

## Does Recombination Change the Picture?

The implication so far is that an evolutionary problem phrased in terms of binary genotypes of length  $L$  should expect to need a minimum of  $L$  generations, adjusted by two estimated fudge factors. The first such factor significantly increases the generations needed; the role of a relatively small population and the loss of information due to mutations incurs considerable costs. The second such factor, arising from an assessment of the likely degree of genetic redundancy, may partially compensate by tending to decrease the generations needed. So far this assessment is based on an asexual GA.



But GAs are typically sexual, and many will expect the addition of recombination to provide a significant speed-up. Indeed, under some circumstances it definitely will.

Mackay (1999) argues that, under particular assumptions, recombination speeds up the asexual rate of 1 bit per generation (where he agrees with the argument above) to something of order  $\sqrt{L}$  bits per generation, where  $L$  is the length of a binary genotype. This is a massive speedup, and simulations based on those assumptions broadly confirm experimentally this factor that Mackay derives analytically.

But these assumptions are completely unrealistic for any problem where GAs are likely to be used. He assumes that fitness is a strictly additive trait; i.e. no epistasis with each locus on the genotype contributing fitness independently of the values at other loci. This can be considered a 'Mt. Fuji' fitness landscape; for such simple scenarios there are much more effective search algorithms than any standard GA.

For instance, consider a population of  $L$  clones of some randomly chosen genotype. Produce  $L$  new offspring, where the  $i^{\text{th}}$  one differs from the parent at the  $i^{\text{th}}$  locus, and compare the fitness of each one with its parent. This immediately identifies the fittest allele at each locus, and allows you to identify the peak of Mt. Fuji in a single generation. No real world GA application comes anywhere close to corresponding to such a simplistic fitness landscape, yet such assumptions of additive fitness are quite often used when biologists do theoretical population genetics. Why is this?

**Micro-evolution and Macro-evolution.** I believe the answer is that biologists very commonly consider issues of *micro-evolution*: they consider an already-evolved species of complex organisms and its ongoing 'struggle' to maintain its fitness through natural selection in the context of continual environmental and competitive challenges that tend to decrease that fitness. Loosely speaking, the fitness of any species, whether single celled amoeba or far more complex mammal, must be around unity if it neither goes extinct nor increases without limit. But this is like maintaining height through constantly climbing up (via natural selection) a downward escalator driven down by 'deterioration of the environment' in its broadest sense. This is the picture given by Fisher's Fundamental Theorem (Fisher, 1958), especially as clarified by Price (1972).

In such a scenario, where any evolved phenotypic trait probably depends on the sum of a number of genetic contributions, then the micro-evolutionary recovery from environmental deterioration may well justify the biologists' assumption of additive fitness. But the scenario faced by those using evolutionary algorithms is usually very different.

They are typically interested instead in *macro-evolution*: the equivalent of mammal species developing from unicellular species over billion of years. If this is characterized by extended periods of stasis, punctuated by individual innovations, for instance in the context of Neutral Networks (Harvey, 2001), then the additive fitness assumption ceases to make sense. One can agree (Ochoa and Harvey, 1999) that the more correlated a fitness landscape is, the more advantageous it is to add recombination. One can agree that recombination is still worth adding to a GA; its role appears largely to be mitigating the negative effects of mutation. But unless one buys into the wholly unrealistic assumptions of Mackay (1999) and others, in the absence of compelling evidence to

the contrary it seems prudent to assume that macro-evolution has the speed limit associated with an asexual GA.

## The Rule of Thumb

Let us first assume binary genotypes, and some standard GA with the equivalent of 50% truncation selection, recombination and appropriate rates (Harvey, 2001) of mutation. If your search-space corresponds to a 20-bit genotype, then 20 generations (at 1 bit per generation) is your starting point. This may be reduced by an estimate of the redundancy; if one plausibly thought that phenotype space partitioned into say  $2^{10}$  equivalence classes, this reduces the figure to 10 generations.

But then very significant allowance must be made for the relatively small population you are likely to be using, and the loss of information due to mutations. One order of magnitude would seem conservative for this, and I tentatively allow 2 orders of magnitude. I.e., for a 20-bit problem with 50% redundancy, I would recommend  $20 \times 0.5 \times 100 = 1000$  generations. For a steady-state GA with population size  $P$ , this is equivalent to  $1000 \times P$  new individuals.

**Real-valued genotypes.** When evolving (e.g.) neural networks, the weights may be encoded in doubles or floats rather than bits on the genotype. As a first approximation, decide whether e.g. 4-bit or 6-bit precision is likely to be adequate for the job, and this gives a direct conversion to the binary case. With a feed-forward network having, say, 3 nodes in the hidden layer, one should note that any perfect network is functionally equivalent with any relabeling of the hidden nodes, and this would give an immediate redundancy factor of  $3! = 6$ , and all such redundancy is helpful.

**Caution.** This heuristic suffers from many imperfections, from lack of knowledge of the details of any particular case, and from the need to make imprecise estimates. It should be treated with caution. Nevertheless, until more principled estimates are proposed it seems better than nothing.

## References

- Fisher, R. A. (1958). *The Genetical Theory of Natural Selection*. 2<sup>nd</sup> ed. New York: Dover Publications.
- Harvey, I. (2001). Artificial evolution: a continuing SAGA. In Gomi, T., editor, *Evolutionary Robotics: From Intelligent Robots to Artificial Life*, LNCS 2217, pages 94-109. Springer.
- Harvey, I. (2011). The microbial genetic algorithm. In Kampis, G., Karsai, I. and Szathmáry, E., editors, *Advances in Artificial Life: ECAL 2009*, Part II, LNCS 5778, pages 126-133. Springer.
- Kimura, M. (1961). Natural selection as the process of accumulating genetic information in adaptive evolution. *Genetical Research*, 2(1):127-140.
- Mackay, D. J. C. (1999). Rate of information acquisition by a species subjected to natural selection. Available online at [www.inference.phy.cam.ac.uk/mackay/gene.pdf](http://www.inference.phy.cam.ac.uk/mackay/gene.pdf)
- Ochoa, G. and Harvey, I. (1999). Recombination and error thresholds in finite populations. In Banzhaf, W. and Reeves, C., editors, *Foundations of Genetic Algorithms 5*, pages 245-264. Morgan Kaufman, San Francisco, CA.
- Price, G. R. (1972). Fisher's 'fundamental theorem' made clear. *Ann. Hum. Genet. Lond.*, 36:129-140.
- Worden, R. P. (1995). A speed limit for evolution. *J. Theor. Biol.*, 176(1):137-152.

# An Emergent Model for Mimicking Human Neuronal Pathways in Silico

Önder Gürcan<sup>1,2</sup>

<sup>1</sup>Ege University, Computer Engineering Department, Üniversite cad. 35100 Bornova, Izmir, Turkey

<sup>2</sup>Paul Sabatier University, Institut de Recherche en Informatique de Toulouse, 118 route de Narbonne, 31062 Toulouse, France  
 onder.gurcan@{ege.edu.tr,irit.fr}

In this study, our aim is to mimic human neuronal pathways without assuming the transition from microscopic to macroscopic scales depend upon mathematical arguments. Human neuronal pathways are natural *complex systems* in which large sets of neurons interact locally and give bottom-up rise to collective macroscopic behaviors. In this sense, correct knowledge of the synaptic effective connections between neurons is a key prerequisite for relating them to the operation of their central nervous system (CNS). However, estimating these effective connections between neurons in the human CNS poses a great challenge since direct recordings are impossible. Consequently, the network between human neurons is often expressed as a *black box* and the properties of connections between neurons are estimated using indirect methods (Türker and Powers, 2005). In indirect methods a particular receptor system is stimulated and the responses of neurons that are affected by the stimulus recorded to estimate the properties of the circuit. However, these neuronal circuits in human subjects are only estimations and their existence cannot be directly proven. Furthermore, there is no satisfactory theory on how these unknown parts of the CNS operate.

We propose a computational *emergent* model<sup>1</sup> that integrates the knowledge from neuroscience and *artificial self-organization* to derive from it the fundamental principles that govern CNS function and its simulation, and ultimately, to reconstruct the human CNS pathways in silico. The term *artificial self-organization* refers to a process enabling a software system to dynamically alter its internal organization (structure and functionality) during its execution time without any explicit external directing mechanism (Serugendo et al., 2011). Our *emergent* model uses temporal data collected from human subjects as an *emergent* macro-level description of the underlying neuronal pathway. Dynamic activity and spiking are modeled at the individual neuron scale. Consequently, the local information in the model is the knowledge about the behavior of individual neurons, such as generation of spikes and transmission of these spikes

to their postsynaptic neurons. The effect of a spike on a target neuron is defined as a temporal membrane potential change in response to the influence of a source neuron that connects to it. That influence is not instantaneous, and is delayed by the physical distance between neurons. However, the interactions of neurons that result in macro-level *emergent* behaviors are unknown and obviously neurons alone are not able to deal with this information. Thus, we defined *adaptive* mechanisms for individual neurons based on biological knowledge. Moreover, to be able to specify purely local information about the reference macro-level pattern, we used the peristimulus frequency (PSF) analysis method (Türker and Powers, 2005).

We model the neuronal network as a dynamic directed graph  $\mathcal{G}(t) = (\mathcal{N}(t), \mathcal{S}(t))$  where  $\mathcal{N}(t)$  denotes the time varying *cooperative* neuron agent (vertex) set and  $\mathcal{S}(t)$  denotes the time varying synapse (edge) set. The set of excitatory (resp. inhibitory) neuron agents at time  $t$  is denoted by  $\mathcal{N}^+(t)$  (resp.  $\mathcal{N}^-(t)$ ) where  $\mathcal{N}(t) = \mathcal{N}^+(t) \cup \mathcal{N}^-(t)$ . A synapse  $\{n, m\}$  delivers spikes from  $n$  to  $m$  with a delay of  $d_{nm}$  and with a synaptic strength of  $\eta$ . We denote the set of postsynaptic neighbors of a neuron agent  $n \in \mathcal{N}$  at time  $t$  as  $Post_n(t) = \{k \in \mathcal{N}(t) | \{n, k\} \in \mathcal{S}(t)\}$  and the set of temporally closest presynaptic neighbors as  $Temp_n(t)$ . The nominal behavior of neuron agents is *spike firing*. The *adaptive* behaviors of neuron agents are subjected to the non-cooperative situations of the agents which are propagated by the feedbacks (Algorithm 1). We denote the set of feedbacks as  $\mathcal{F}$  and we model sending a feedback  $f_a \in \mathcal{F}$  using an action of the form  $send(f_a, \mathcal{R})$  where  $a$  is the source of  $f$  and  $\mathcal{R} \subset \mathcal{N} \setminus \{a\}$  is the set of receiver agents. The *tuning behavior* of neuron agents is modelled using an action of the form  $tune(\{n, m\}, f)$  for  $n, m \in \mathcal{N}(t)$  and  $f \in \mathcal{F}$ , which correspond to the adjustment of  $\{n, m\}.\eta$  by  $f$  at time  $t$ . The *reorganization behaviors* of neuron agents are modeled using actions of the form  $add(\{n, m\})$  and  $remove(\{n, m\})$  for  $n, m \in \mathcal{N}(t)$ , which correspond to the formation and suppression (respectively) of  $\{n, m\}$  at time  $t$ . The *evolution behaviors* are modelled using actions of the form  $create(n, m)$ ,  $createInverse(n, m)$  and  $remove(n)$

<sup>1</sup>This model has been recently reported in (Gürcan et al., 2013, 2012).

for  $n, m \in \mathcal{N}(t)$ .  $\text{create}(n, m)$  corresponds to the creation of a neuron agent between  $n$  and  $m$  by  $n$  having the same type of  $n$ .  $\text{createInverse}(n, m)$  corresponds to the creation of a neuron agent between  $n$  and  $m$  by  $n$  having the opposite type of  $n$ .  $\text{remove}(n)$  corresponds to the suppression of the neuron agent  $n$  by itself.

**Algorithm 1** Response to the feedback  $f_m$  received at time  $t$  of neuron  $n$  where  $f_m \in \mathcal{F}_{if}$  and  $m, n \in \mathcal{N}(t)$ .

---

```

1:  $\square$  tuning condition
2:   if  $m \in \text{Post}_n$  then tune( $\{n, m\}.\eta, f_m$ )
3:  $\square$  reorganization condition
4:   if  $m \in \text{Post}_n$  then
5:     if  $(n \in \mathcal{N}^+ \wedge f_m \downarrow) \vee (n \in \mathcal{N}^- \wedge f_m \uparrow)$  then
6:       remove( $\{n, m\}$ )
7:     else //  $m \notin \text{Post}_n$ 
8:       if  $(n \in \mathcal{N}^+ \wedge f_m \uparrow) \vee (n \in \mathcal{N}^- \wedge f_m \downarrow)$  then
9:         add( $\{n, m\}$ )
10:  $\square$  evolution condition
11:   if  $m \in \text{Post}_n$  then
12:     if  $(n \in \mathcal{N}^+ \wedge f_m \uparrow) \vee (n \in \mathcal{N}^- \wedge f_m \downarrow)$  then
13:       create( $\text{Pre}_n, m$ )
14:     else //  $m \notin \text{Post}_n$ 
15:       if  $(n \in \mathcal{N}^+ \wedge f_m \downarrow) \vee (n \in \mathcal{N}^- \wedge f_m \uparrow)$  then
16:         createInverse( $\text{Pre}_n, m$ )
17:  $\square$  propagation condition
18:   send( $f_m, \text{Temp}_n(t)$ )

```

---

We applied the model to the reflex responses of single motor units obtained from conscious human subjects. The exact information we used about the underlying pathways is that *sensory neurons make monosynaptic connections with the alpha motoneuron*. In this sense, we considered this path as the shortest path in the underlying network and defined its duration as  $l$ . Therefore, we initialized the simulations as  $\mathcal{N}(0) = \{s, m\}$ ,  $\mathcal{S}(0) = \{\{s, m\}, \{m, \emptyset\}\}$  and  $d_{sm} = d_{m\emptyset} = l/2$  where  $s, m \in \mathcal{N}^+$  and  $l$  is the latency of the estimated beginning of the pathway.

The results show that (e.g., Figure 1), **the emergent neuronal network model learns to generate what is observed in human subjects in cellular resolution**. What makes the model promising is the fact that, to the best of our knowledge, it is the first realistic model to self-wire an artificial neuronal network by efficiently combining neuroscience with artificial self-organization. Although there is no evidence yet of the model's connectivity mapping onto the human connectivity, we anticipate this model will help neuroscientists to learn much more about human neuronal networks, and could also be used for predicting hypotheses to lead future experiments.

## References

Gürçan, Ö., Bernon, C., Türker, K. S., Mano, J.-P., Glize, P., and Dikenelli, O. (2012). Simulating human single motor

units using self-organizing agents. In *Self-Adaptive and Self-Organizing Systems (SASO), 2012 IEEE Sixth International Conference on*, pages 11–20.

Gürçan, Ö., Türker, K. S., Mano, J.-P., Bernon, C., Glize, P., and Dikenelli, O. (2013). Mimicking human neuronal pathways in silico: an emergent model on the effective connectivity. *Journal of Computational Neuroscience* (accepted).

Serugendo, G., Gleizes, M.-P., and Karageorgos, A., editors (2011). *Natural Computing Series*. Springer.

Türker, K. S. and Powers, R. K. (2005). Black box revisited: a technique for estimating postsynaptic potentials in neurons. *Trends in neurosci.*, 28(7):379–386.

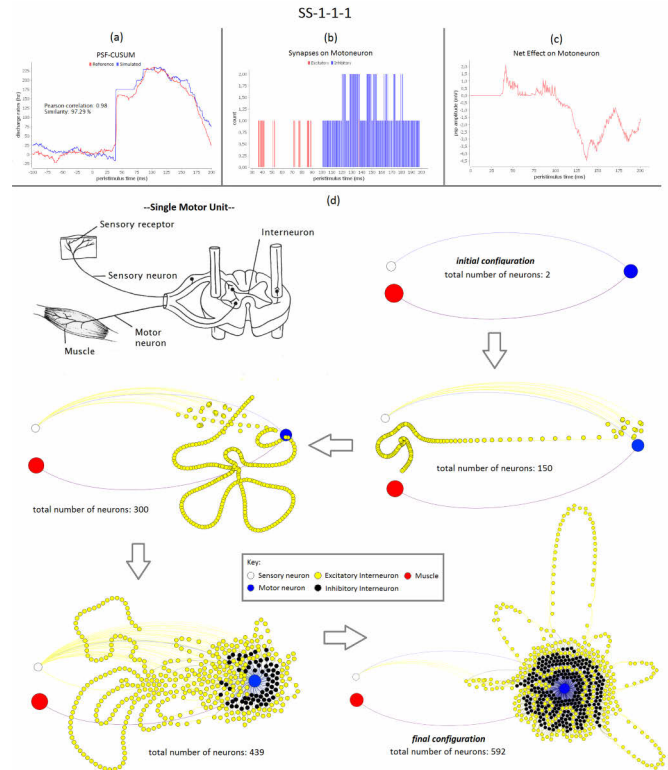


Figure 1: The results that came out of a simulation run at the end of its effort to learn the global pattern obtained from the human reflex experiment SS-1-1-1 (Gürçan et al., 2013). (a) PSF-CUSUM diagrams of the reference data (red line) and its simulated replication (blue line). Pearson-correlation of these lines is 0.98 and thus their similarity is 97.29%. (b) The temporal distribution of created excitatory (red) and inhibitory (blue) synapses on the motoneuron. (c) The net PSP on motoneuron caused by its presynaptic connections given in (b). (d) The cinematic representation of the evolution of the neural network from the initial configuration towards the final configuration together with the number and the sign of neurons. Note also that, in the final configuration, the extent of the pathways that represent the long latency reflex responses are emerging as neuronal loops.

# Effects of inducer intake kinetics on the dynamics of gene expression

Huy Tran and Andre S. Ribeiro

Dept. of Signal Processing, Tampere University of Technology, P.O Box 553, 33101 Tampere, Finland  
andre.ribeiro@tut.fi

## Abstract

Many genes are only active following the intake of an inducer by the cell, via passive diffusive, positive, or negative feedback intake mechanisms. Based on measurements of the *in vivo* kinetics of intake and subsequent transcription events in *Escherichia coli*, we use stochastic models to investigate how the kinetics of intake affects both transient and near-equilibrium dynamics of gene expression. We find that the intake kinetics affects mean and variability of the transient time to reach the steady state of proteins numbers and also the degree of fluctuations in these numbers. Fluctuations in the extracellular number of inducers affects the variability of protein numbers at steady state in a degree that differs with the intake kinetics. Finally, changing the intake kinetics of an inducer of a genetic switch allows tuning the bias in the choice of noisy attractor. We conclude that the kinetics of inducer intake affects transient and near-equilibrium gene expression dynamics and, consequently, the phenotypic diversity of organisms in fluctuating environments.

## Introduction

To survive, cells must adapt to environmental changes, such as in the concentration of nutrients and toxics. Some of these changes can occur at rates faster than, e.g., the cell cycle, and thus require rapid adaptability from the cells. This adaptability may involve modifications in the kinetics of membrane-associated mechanisms (Sajbidor, 1997), metabolic rates (Talwalkar and Kailasapathy, 2003), or gene expression (Yamamoto and Ishihama, 2005; Allen and Tresini, 2000).

Studies suggest that organisms such as *Escherichia coli* can adjust the reception of some external signals. For example, in normal conditions, transcription of the genes of the *lac* operon (Elf et al., 2007; Hansen et al., 1998) is inhibited by the native *lac* repressor. When allolactose is present in the environment, it is absorbed by passive intake transport and triggers the expression of the *lac* genes. One of the proteins expressed, *lacY*, enhances the intake of allolactose further, thus forming a positive feedback. Such feedback mechanisms are particularly useful in saving cellular resources in periods when inducers are not present (Jacob and Monod, 1961). On the other hand, negative feedback

mechanisms are appropriate for, e.g., quickly pumping unwanted substances out of the cell (Schnappinger and Hillen, 1996). One example is the tetracycline intake system, as the tetracycline-induced proteins *tetA* actively transport the tetracycline out of cell (Beck et al., 1982).

A recent study (Megerle et al., 2008) showed that the timing of intake of inducers differs widely between cells in monoclonal populations of *E. coli*. Such variability in intake times was visible in the differing timings for the appearance of proteins in cells following the introduction of the inducer in the media even though, after a transient period, all cells exhibited the same rate of protein production. Another recent study (Makela et al., 2013) supported this hypothesis, by showing that there is a wide variability in the timing of activation of transcription following the appearance of the inducer in the media that is not due to noise in gene expression but causes high cell to cell variability in RNA numbers for long periods of time (i.e. longer than several cell cycles). This source of phenotypic diversity is likely to be of particular relevance in fluctuating environments (Acar et al., 2013; Ribeiro, 2008).

Here, using detailed stochastic models of gene expression and intake processes in *E. coli*, we investigate if differing intake kinetics results in differing kinetics of expression of the target gene both in the transient period for protein numbers to reach near-equilibrium, as well as in the subsequent stable phase. Next, we investigate how fluctuations in inducer numbers in the environment affect this variability. Finally, we investigate whether the intake kinetics of inducers can affect the behavior of a small genetic circuit, namely, a toggle switch.

## Methods

We compare the kinetics of expression when the inducer of the target gene enters cells via positive feedback mechanisms, passive diffusion, or negative feedback mechanisms. While varying the intake kinetics, the mean number of proteins expressed by the target gene is kept invariant in the stable phase.

The dynamics of the models is driven by the delayed



Stochastic Simulation Algorithm (delayed SSA) (Roussel and Zhu, 2006). This algorithm, unlike the original SSA (Gillespie, 1977), allows delaying the release of products, following a reaction. Furthermore, it differs from previous algorithms that can accommodate delays (e.g. (Bratsun et al., 2005; Barrio et al., 2006)) in that it can handle multiple delayed events in one reacting event, which facilitates the modeling of genetic circuits (Ribeiro, 2010). The delayed SSA uses a wait list to store delayed output events. The wait list is a list of elements (e.g., proteins being produced), each to be released after a time interval has elapsed (also stored in the wait list).

Each model includes an extracellular environment, which contains inducers, a cellular intake mechanism of inducers, and a gene expression mechanism that requires activation. The proteins produced can affect the intake kinetics, so as to model positive or negative feedback mechanisms. The models are simulated by SGNSim (Ribeiro and Lloyd-Price, 2007). All parameter values are extracted from measurements in *E. coli*, unless stated otherwise.

### Environment and passive transport of inducers

We assume that the inducers in the environment are inexhaustible. To model passive diffusion of inducers into cells, we set a rate constant of intake ( $k_{I_{in}}$ ) and an extracellular amount of inducers ( $I_e$ ) equal to 1, for simplicity. The intake is modeled by the following reaction:



where  $I$  is the number of inducers inside a cell. Note that, in all cases, even when an active mechanism is present, there is always passive intake.

To model fluctuations in inducer numbers in the environment, we assume that these follow a Gaussian distribution with standard deviation  $\sigma_e$  and unity mean. The fluctuations are set to occur at a rate slow enough to allow for one fluctuation to have a visible effect in the protein numbers before the next one occurs. For that, we use a first order autoregressive model to restrict the degree of change in inducer numbers from one moment to the next, with the following update rule:

$$I_e(t) = 1 - \phi + I_e(t - \delta t) * \phi + \epsilon_t \quad (2)$$

where  $\phi$  is constant,  $\epsilon_t$  is white noise with standard deviation of  $\sigma_t$ , and  $\delta t$  is the update interval. The model generates values for the extracellular inducer numbers according to  $I_e \sim N(1, \sigma_e^2)$ , where:

$$\sigma_e^2 = \frac{\sigma_t^2}{1 - \phi^2} \quad (3)$$

The inducers' extracellular concentration thus has the autocorrelation function's decay rate of  $-\ln(\phi)/\delta t$ . By tuning

$\phi$ , one can adjust the rate of change in this concentration. Finally, cells can dispose of inducers via diffusion, modeled as a first order reaction event:

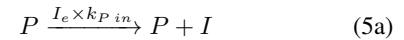


Reaction 4 is assumed to account also for possible degradation events of inducers when inside the cell.

### Active transport mechanisms

We assume that the active transport rate is proportional to the number of proteins of the target gene. Let such transport be done by a protein  $P$ .

Positive feedback mechanisms are modeled by reaction 5a, where one inducer  $I$  is introduced in the cell by a protein  $P$ , while negative feedback mechanisms are modeled by reaction 5b, where one inducer is pumped out of the cell by a protein  $P$ :



These two mechanisms are never present simultaneously.

### Gene expression

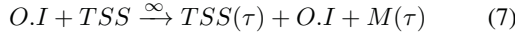
We assume that the gene only expresses once activated. An inducer  $I$  interacts with the operator site  $O$  at the promoter region of the gene via the following reactions:



where  $k_a$  and  $k_d$  are the association and the disassociation rate constants, respectively. We assume half lives of inducers much longer than the expected time for disassociation to occur. Thus, we do not model degradation of inducers when bound to the operator. Additionally, we assume that leaky expression is negligible and that the operator site is not overlapped by the RNA polymerase for any significant time, so that the interaction between promoter and inducer is independent of the transcription process, particularly initiation.

The model of gene expression used was proposed and validated in (Ribeiro et al., 2006; Zhu et al., 2007), by comparing its kinetics with the real-time production of tsr-venus proteins under the control of a lac promoter in *E. coli* (Yu et al., 2006). The model consists of transcription (7), in this case of the activated gene, and translation of the resulting RNA molecules (8). Also modeled are first order degradation processes of RNA (9) (Bernstein et al., 2002) and proteins (10). Transcription events can occur when the operator-inducer complex  $O.I$  is formed. RNA polymerases are not explicitly modeled, as it is assumed that these exist in sufficient amount so that fluctuations in their numbers are not

significant. The transcription start site (TSS) is modeled explicitly so that transcription initiation events do not interfere with operator-activator reactions:



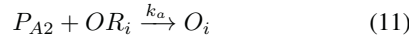
Reaction 7 describes the process of transcription. In particular,  $\tau$  (which follows a Gamma distribution (Kandhavelu et al., 2011)) accounts for the finding of a promoter region by an RNA polymerase, the formation of the closed complex at the transcription start site, the open complex formation, and finally, the promoter escape (DeHaseth et al., 1998) and elongation. Of these, in general, the most rate limiting steps are the isomerization steps and the open complex formation (McClure, 1985; Lutz et al., 2001). To model this multi-step process, we set the reaction rate to infinity, which causes the reaction to occur the moment the reactants become available. Thus, in this model,  $\tau$  determines the interval between consecutive productions of transcripts when the gene is induced.

Also, given the short duration of the elongation time in comparison to transcription initiation (Kandhavelu et al., 2011), the transcript ( $M$ ) is released at the same time as the TSS (i.e. the elongation time is assumed negligible). This allows for translation events of the RNA to initiate (reaction 8) as soon as the assembly of that RNA begins (Miller et al., 1970).

### Genetic Toggle Switch

The toggle switch consists of a network of two genes (here, B and C), whose proteins ( $P_B$  and  $P_C$ , respectively) repress the other gene's activity. We assume that these two genes are only active when bound by a protein produced by an operon, 'A', which is itself activated by the extracellular inducer. The transcription and translation processes in each of these genes are modeled as described in the previous section.

In this model, the operon A, once activated by the inducer, expresses two proteins,  $P_{A1}$  and  $P_{A2}$ . The former is involved in the intake of the inducer via a feedback mechanism (described in a previous section), while the latter activates genes B and C via the following reaction:



where  $P_{A2}$  is the activator,  $OR_i$  is an operator site at the promoter of either gene B or C in the inactive state ( $i = B, C$ ),  $k_a$  is the association rate constant, and  $O_i$  is the operator region with the activator bound to it.

The interactions between genes B and C form a switch. Namely, each of these genes, once activated by  $P_{A1}$ , is free

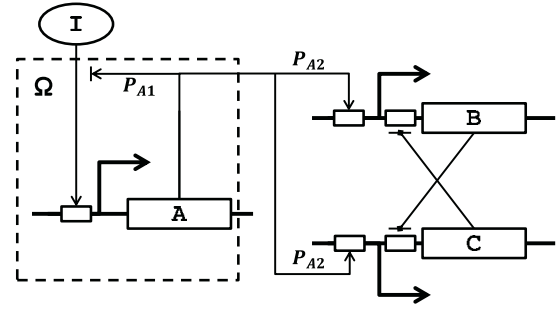
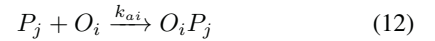


Figure 1: Model switch and activation mechanism. Inducers enter the cell by the intake mechanism, whose kinetics is determined by  $\Omega$  (dashed box). Protein  $P_{A1}$  is responsible for the feedback mechanism, while  $P_{A2}$  activates genes B and C, whose mutual interactions form a switch.

to express or to be repressed by the protein of the other gene ( $P_B$  represses gene C while  $P_C$  represses gene B). Such repressions occur at a second operator site at the promoter regions, via:



where  $P_j$  is the repressor,  $O_i$  is an active operator site of either gene B or C,  $O_i P_j$  is the operator region with a repressor bound to it, and  $k_{ai}$  is the association rate constant of that repressor. Importantly, this rate differs in the two genes, being higher for gene C, which biases the choice of noisy attractor made by the switch, when first initialized (Ribeiro and Kauffman, 2007). The noisy attractor favored is “gene C on and gene B off”. In Fig. 1 we show a schematic representation of this model.

### Characterization of the intake process

To compare the effects of different active intake systems we modeled these such that, for the same extracellular concentration of inducers, one has the same mean protein number of the inducible gene at steady state ( $[P]$ ), for varying active transport kinetics. To achieve this, the mean number of inducers in the cells at steady state ( $[I]$ ) has to be kept constant for varying intake kinetics, which is done by tuning the kinetics of passive intake.

In the stable phase, the influx and outflux rates of inducers are identical. Let  $f_1$  and  $f_2$  be the passive influx and outflux of inducers, respectively. These occur, respectively, via reactions 1 and 4. Given these,  $f_1 = I_e \times k_{I in}$  while  $f_2 = [I]k_{I out}$ . Let  $f_3$  be the flux due to the active transport. Since the flux ( $f$ ) from active and passive transports, in the stable phase, must equal zero, then:

$$\begin{aligned} f &= f_1 - f_2 + f_3 \\ &= [I_e]k_{I in} - [I]k_{I out} + f_3 = 0 \end{aligned} \quad (13)$$

The active transport flux  $f_3$  can thus be calculated as follows, in each condition:

$$f_3 = \begin{cases} [I_e][P]k_{P\text{ in}} > 0 & \text{positive feedback} \\ -[P][I]k_{P\text{ out}} < 0 & \text{negative feedback} \\ 0 & \text{passive diffusion} \end{cases} \quad (14)$$

Finally, we define  $\Omega$  as the log ratio between the passive disposal flux and the passive intake flux. The value of  $\Omega$  informs on the rate at which inducers are transported into cell by passive intake when in the stable phase:

$$\Omega = \lg\left(\frac{f_2}{f_1}\right) = \lg\left(\frac{f_3}{f_1} + 1\right) = \lg\left(\frac{[I]k_{I\text{out}}}{k_{I\text{in}}}\right) \quad (15)$$

$\Omega$  can take positive values in the presence of positive feedback (i.e.  $f_3 > 0$ ), and negative values in the presence of negative feedback ( $f_3 < 0$ ). Given passive intake alone ( $f_3 = 0$ ),  $\Omega = 0$ .

## Results

The models used to study the effects of the kinetics of the intake of inducers on the dynamics of gene expression are stochastic. Thus, to assert if their dynamics changes as a function of a parameter's value, we perform tests of statistical significance. Also, the models are initialized without any proteins of the target gene, so as to assess the kinetics both at the stable phase, as well as during the transient to reach the stable phase.

### Gene expression and intake kinetics

We first study the dynamics of protein numbers of a target gene as a function of the intake kinetics of the inducer, both in the transient phase and in the near-equilibrium or stable phase. In all cases, we use the following parameter values for the model of gene expression:  $k_P = 0.005 \text{ s}^{-1}$  (Taniguchi et al., 2010),  $d_M = 0.002 \text{ s}^{-1}$  (Bernstein et al., 2002), and  $d_P = 0.0005 \text{ s}^{-1}$  (Taniguchi et al., 2010). Also, we let  $\tau$  be a random variable following a gamma distribution  $\Gamma(\alpha, \theta)$ , with the shape  $\alpha$  equal to 2 (Kandhavelu et al., 2011) and the scale  $\theta$  equal to 25 s. The value of  $\theta$  was set so that, given the other parameter values, the mean RNA number in the stable phase is  $\sim 10$ , in accordance with *in vivo* measurements in *E. coli* (Taniguchi et al., 2010). The reaction rates of inducer-operator interactions are set to:  $k_a = 10^{-5} \text{ s}^{-1}$  and  $k_d = 0.02$ , so that the expected time for inducers to bind to the promoter is in accordance with measurements reported in (Elf et al., 2007).

We vary  $\Omega$ , while maintaining constant the mean protein ( $[P] \sim 30$ ) and mean inducer numbers within the cells ( $[I] \sim 1400$ ), when in the stable phase. For this, the range of variation of  $\Omega$  was constrained between -1 and 1. This range complies with measurements of the intake kinetics of known inducers (namely, of tet and lacY) in *E. coli* (Brown and Hogg, 1972; Hansen et al., 1998; Beck et al., 1982).

For each value of  $\Omega$ , we simulate 500 cells, each for 60 000 s, sampling their state every 60 seconds. We define the stable phase as the phase in which the mean protein numbers in the cells do not differ, in a statistical sense, for different values of  $\Omega$ . We found that all cells reach the stable phase after, at most,  $t = 4 \times 10^4 \text{ s}$ . In Table 1, we show the p-values of the Kolmogorov-Smirnov (KS) test comparing the distribution of mean protein numbers when  $\Omega = 0$  (passive intake) with each of these distributions for the other values of  $\Omega$ , in the stable phase.

Table 1: p-values of the KS test comparing the distribution of mean protein numbers when  $\Omega = 0$  and when  $\Omega$  takes other values, in the stable phase ( $t > 4 \times 10^4 \text{ s}$ )

| $\Omega$ | -1   | -0.4 | 1    | 0.4  | 1    |
|----------|------|------|------|------|------|
| p-value  | 0.03 | 0.19 | 1.00 | 0.26 | 0.34 |

The p-values from likelihood ratio tests with the null hypothesis that the distributions are identical are larger than 0.01, thus, we cannot reject that they are identical in a statistical sense. Therefore, in this range of values of  $\Omega$ , the models have identical mean protein numbers over time, in a statistical sense, when in the stable phase.

We next study the intracellular dynamics of inducer numbers as a function of  $\Omega$ . In Fig. 2a we show the mean number of inducers in the cells over time, from the start of the simulations. These vary significantly as a function of  $\Omega$ , before reaching the stable phase. In particular, for  $\Omega < 0$  (negative feedback), there is a rapid influx of inducers, followed by a steady decrease towards the numbers at near-equilibrium. For  $\Omega > 0$  (positive feedback), the inducer numbers take longer time to reach near-equilibrium. The passive intake mechanism ( $\Omega = 0$ ) is, of the cases modeled, the one for which the intracellular inducer numbers stabilizes faster. We found by inspection that this mean time is minimized for values of  $\Omega$  close to, but slightly smaller than 0. Finally, from Fig. 2b we observe that the proteins reach slower the numbers observed in the stable phase the greater is  $\Omega$ .

In Fig. 3, we show the mean transient time ( $t_0$ ) for each value of  $\Omega$ , along with the square of the coefficient of variation ( $CV^2(t_0)$ ), obtained from the multiple simulations in each condition. We find that the mean  $t_0$  is shorter for negative intake mechanisms. However, the  $CV^2(t_0)$  does not change significantly with  $\Omega$ .

Next, we assess the fluctuations in protein and inducer numbers in the stable phase as a function of  $\Omega$ . Fig. 4 shows the variance over the mean ( $\sigma^2/\mu$ ) ratio (i.e. the fano factor) of these numbers. This quantity is minimized for  $\Omega = 0$  in the case of intracellular inducer numbers. In the absence of feedback, these follow a Poisson distribution, as expected since both the passive intake and disposal are first-order processes. When there is an active feedback mechanism, the noise in protein numbers causes an noise in the intracellular

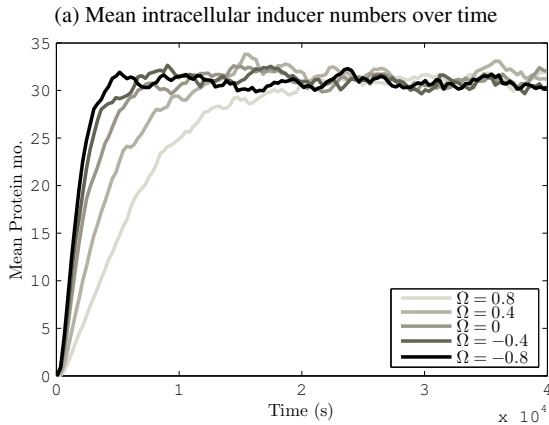
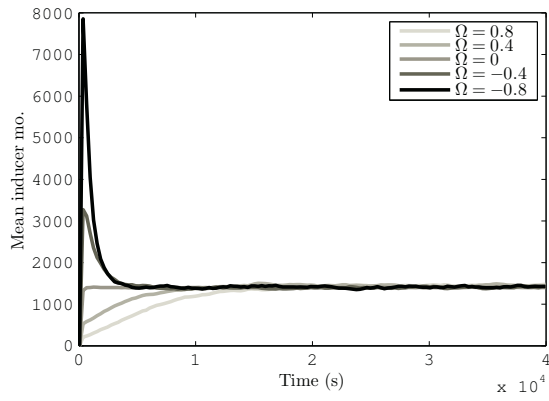


Figure 2: Mean numbers of inducers (top) and proteins (bottom) over time:  $\Omega < 0$  corresponds to negative feedback,  $\Omega = 0$  corresponds to passive intake, and  $\Omega > 0$  corresponds to positive feedback mechanisms.

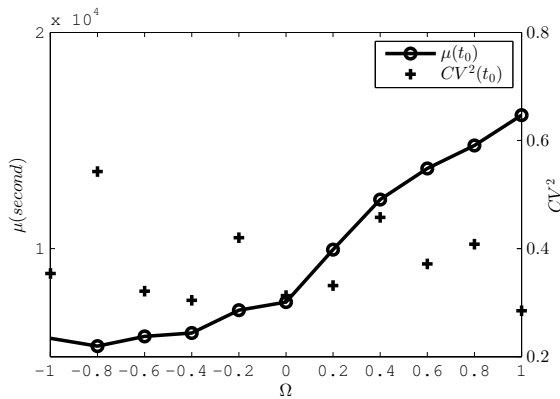


Figure 3: Mean ( $\mu$ ) and  $CV^2$  of the transient time,  $t_0$ , for protein numbers to reach near-equilibrium as a function of  $\Omega$ .

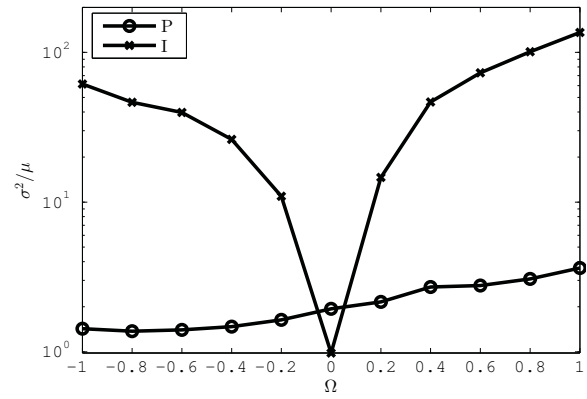


Figure 4: Noise in protein and intracellular inducer numbers in the stable phase as a function of  $\Omega$ .

inducer numbers to be higher, then when only passive intake is present. The stronger the feedback mechanism (larger deviation from  $\Omega = 0$ ), the stronger is this effect.

On the other hand, the noise in protein numbers increases for increasing  $\Omega$ , being lower for negative feedback mechanisms and higher for positive feedback mechanisms. To investigate this, we calculated the normalized cross-correlation between protein and intracellular inducer numbers in the stable phase for varying  $\Omega$  (Fig 5). For positive feedbacks ( $\Omega > 0$ ), the protein and inducer numbers are positively correlated. This means that the noise in intracellular inducer numbers will be propagated to the protein numbers, causing it to be higher than in the passive diffusion case. In the regime of negative feedbacks, the numbers of intracellular inducers and proteins are anti-correlated, and the noise in the numbers of proteins is suppressed, when compared to passive diffusion case. Finally, as expected, in the absence of feedback mechanisms, the noise in intracellular inducer numbers does not affect the protein numbers, which is indicated by zero cross-correlation at  $\Omega = 0$ .

Finally, we study how the intake mechanisms behave in environments with fluctuating number of inducers. We assume that the extracellular number of inducers follows a Gaussian distribution with variance  $\sigma_e^2 = 0.2$  and unity mean, generated by the autoregressive model (see methods). We set  $\delta t$  to 30 s. We set the rate of environmental change,  $\phi$ , from 0.5 to  $\sim 1$ . The closer the value of  $\phi$  to 1, the slower the decay rate of the autocorrelation function of the extracellular inducer concentration. For  $\phi \sim 1$ , the extracellular inducer concentration is constant, corresponding to  $\sigma_e = 0$ . For each pair of values  $[\phi, \Omega]$ , we simulate one cell for  $5 \times 10^6$  s, sampling every 60 s. Fig. 6 shows the changes in fluctuations in protein numbers in the stable phase (as assessed by the  $CV^2$ ) due to the fluctuations in the inducer numbers, relative to when in stable environments.

From Fig. 6, with values of  $\phi$  larger than 0.8, the noise



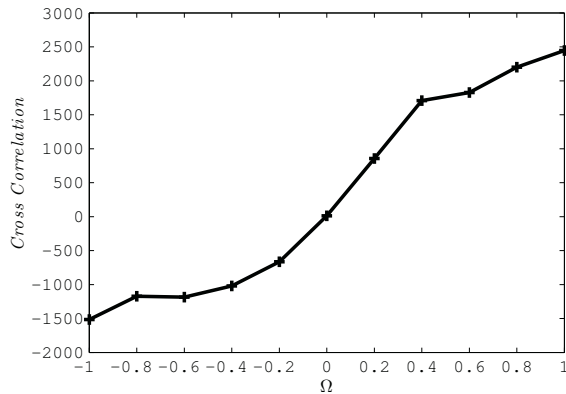


Figure 5: Cross-correlation between protein and inducer numbers in the stable phase.

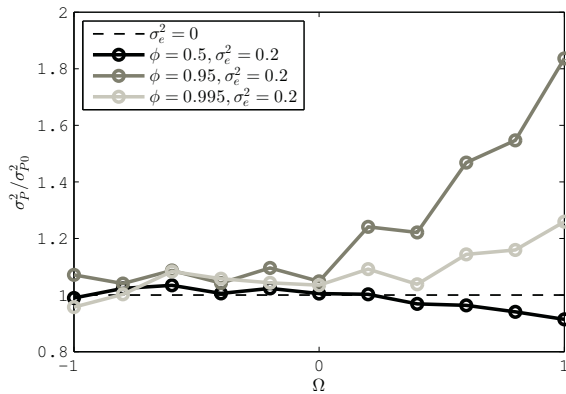


Figure 6: Noise amplification ratio ( $\sigma_P^2/\sigma_{P0}^2$ ) due to fluctuations in the number of inducers in the environment.

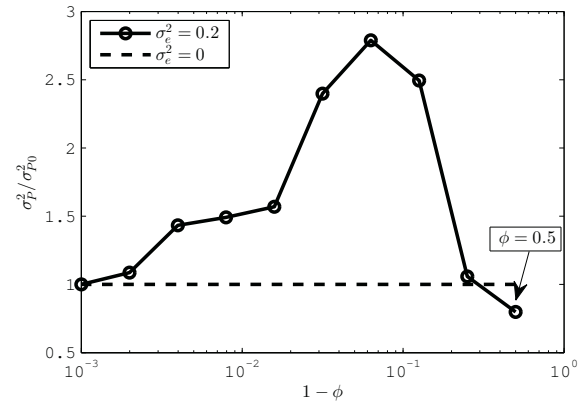


Figure 7: Noise amplification ratio ( $\sigma_P^2/\sigma_{P0}^2$ ) with  $\Omega = 1$ ,  $\sigma_e = 0.2$  for different values of  $\phi$ .

amplification in the protein numbers appears to increase with increasing  $\Omega$ . For each value of  $\phi > 0.8$ , we performed tests of statistical significance between the protein numbers when  $\Omega$  is 0, and when is -1 and 1, respectively. In both tests, we found that the distributions are distinct (p-values smaller than  $10^{-10}$ ), confirming that the increase in the noise amplification effect with increasing  $\Omega$  is statistically significant.

For positive feedback mechanisms, the noise amplification ratio for different values of  $\phi$  resembles a band pass filter (Fig.7). As  $\phi$  increases up to 0.9, the fluctuations in the external inducer numbers propagate more efficiently to the protein numbers of the induced gene (as shown in Samoilov et al. (2002)). When  $\phi$  increases beyond 0.9, the noise amplification ratio decreases as the positive feedback, affected by the extracellular inducer numbers, loses the ability to reflect the fluctuations in protein numbers. Interestingly, for  $\Omega = 1$ ,  $\phi = 0.5$ ,  $\sigma_e = 0.2$ , the protein  $CV^2$  is reduced by  $\sim 10\%$  when compared with the noiseless case (i.e.  $\sigma_e = 0$ ). This reduction is significant, namely the resulting distributions of protein numbers in the two cases are statistically distinct (p-value smaller than  $10^{-10}$ ).

### Inducible genetic switch

We study the behavior of a biased genetic switch as a function of the intake kinetics of an inducer. The model consists of two genes, B and C, which form a switch via mutually repressing interactions, and of a third gene, A, responsible for, once activated by the inducer, activate both genes B and C (figure 1).

In this model,  $P_{A_2}$  activates the expression of genes B and C in the same fashion as reaction 6, with the association rates:  $k_{A_2B} = 0.001 \text{ s}^{-1}$ ,  $k_{A_2C} = 0.002 \text{ s}^{-1}$ . The other parameters of genes B and C are:  $k_{PB} = k_{PC} = 0.05 \text{ s}^{-1}$ ,  $d_{MB} = d_{MC} = 0.002 \text{ s}^{-1}$ ,  $d_{PB} = d_{PC} = 0.0005 \text{ s}^{-1}$ . The formation of the open complex is regulated by:  $\tau_B, \tau_C \sim \Gamma(1, 100 \text{ s})$ . Finally, we denote  $[t_B]$  and  $[t_C]$  as the

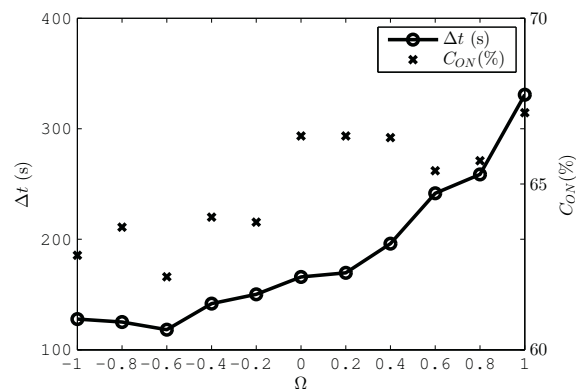


Figure 8: Mean temporal difference between the first activation of genes B and C ( $\Delta t$ ), and bias in the choice of noisy attractor towards “gene C on” ( $C_{ON}$ ).

mean elapsed times from the introduction of inducers to the first closed complex formation in each promoter. The bias in the association rate constants biases the moments of formation of the closed complex. Namely, on average, these occur at moments that differ in time by:  $\Delta t = [t_B] - [t_C] > 0$ .

Given these parameter values, we observed that both noisy attractors of the switch are stable enough so that, once a noisy attractor is reached, the switch will remain there until the end of the simulation. Also, as noted in the methods section, the association rate of the activator protein  $P_{A_2}$  is higher in the case of gene C, which biases the first choice of noisy attractor of the switch. For  $\Omega = 0$ , the first noisy attractor selected by the switch will be “gene C on”  $\sim 65\%$  of the times (see Fig. 8).

We study the bias in the choice of noisy attractor as a function of  $\Omega$ . For each value of  $\Omega$ , we simulate 1000 independent cells, each in 30 000 s, sampled every 5 seconds. The results are shown in Fig. 8. From this figure, for increasing values of  $\Omega$ , the bias in choice of noisy attractor is reduced from  $\sim 67\%$  to  $\sim 62\%$ .

We performed statistical tests of significance comparing the distribution of  $\Delta t$  when  $\Omega = 0$  to the same distribution when  $\Omega = -1$  and 1 respectively. The test results show that all tested pairs of distributions are distinct (p-values smaller than  $10^{-10}$ ).

This result can be explained as follows. As  $\Omega$  is increased,  $P_A$  reaches the stable phase slower. Thus, both  $[t_B], [t_C]$  increase, but not by equal amounts (e.g. for half the number of proteins A, each of these times is doubled). Accordingly,  $\Delta t$  increases and the distribution of chosen noisy attractors becomes more biased.

## Discussion

Many genes in *E. coli*, as well as other single-celled organisms, only become active in response to an external signal,

either individually, or as part of a small network. Additionally, even when active, in the stable phase, most genes exhibit very small mean RNA numbers (from one to a few) (Taniguchi et al., 2010). This implies that differences in the intake time of inducers between sister cells can have significant implications on phenotypic differences between them. Similarly, differences in the kinetics of the intake process of different inducers may lead to significant differences on the mean and variability of response times to those inducers. Relevantly, Recent measurements *in vivo* showed that the intake time of inducers can be of the same order of magnitude as the cell division time and transcription initiation (Kandhavelu et al., 2011). The degree of cell-to-cell variability in these times is also equally high (Megerle et al., 2008; Makela et al., 2013).

Using a stochastic model with parameter values extracted from measurements in *E. coli*, we showed that the nature of the intake mechanism, that is, whether it is based on passive diffusion, positive feedback or negative feedback mechanism, has a significant impact on the dynamics of gene expression, both in the transient phase, as well as in the stable phase. The intake kinetics not only affects mean and variability of the transient time to reach the stable phase but also the degree of fluctuations in these numbers once it that phase. These effects are tangible in the behavior of small genetic circuits.

The results presented here show that the kinetics of the response, in terms of gene expression, of single-celled organisms to external signals, depends to great extent not only on the intake mechanism of the inducer/repressor molecule as well as on the mechanisms of transcription and translation. Also relevant is the observation that the intake mechanism also has an effect on the kinetics of gene expression, long after the transient period. This implies that the kinetics of genes responsive to environmental signals ought to be studied accounting for the effects of the intake mechanism on RNA and protein numbers dynamics. In the future, it would be of interest to further explore how active transport mechanisms, able of positive or negative feedback processes, can be used to tune the behavior and adaptability of small genetic circuits to fluctuating environments.

## Acknowledgements

We acknowledge Tekes and Academy of Finland. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

## References

- Acar, M., Mettatal, J. T., and van Oudenaarden, A. (2013). Stochastic switching as a survival strategy in fluctuating environments. *Nature Gen.*, 40(4):471–475.
- Allen, R. and Tresini, M. (2000). Oxidative stress and gene regulation. *Free Radic. Biol. Med.*, 28(3):463–499.

- Barrio, M., Burrage, K., Leier, A., and Tian, T. (2006). Oscillatory regulation of *hes1*: Discrete stochastic delay modelling and simulation. *PLOS Comp. Biol.*, 2(9):1017–1030.
- Beck, C., Mutzel, R., Barbe, J., and Muller, W. (1982). A multi-functional gene (*tetr*) controls *tn10*-encoded tetracycline resistance. *J. Bacteriol.*, 150(2):633–642.
- Bernstein, J., Khodursky, A., Lin, P., Lin-Chao, S., and Cohen, S. (2002). Global analysis of mrna decay and abundance in *escherichia coli* at single-gene resolution using two-color fluorescent dna microarrays. *Proc. Natl. Acad. Sci. USA.*, 99(15):9697–9702.
- Bratsun, D., Volfson, D., Tsimring, L., and Hasty, J. (2005). Delay-induced stochastic oscillations in gene regulation. *Proc. Natl. Acad. Sci. USA.*, 102(41):14593–14598.
- Brown, C. and Hogg, R. (1972). A second transport system for l-arabinose in *escherichia coli* b/r controlled by the *arac* gene. *J. Bacteriol.*, 111(2):606–613.
- DeHaseth, P. L., Zupancic, M., and M. T. Record, J. (1998). Rna polymerase-promoter interactions: the comings and goings of rna polymerase. *J. Bacteriol.*, 180(12):3019–3025.
- Elf, J., Li, G., and Xie, X. (2007). Probing transcription factor dynamics at the single-molecule level in a living cell. *Science*, 316:1191–1194.
- Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361.
- Hansen, L., Knudsen, S., and Srensen, S. (1998). The effect of the *lacy* gene on the induction of *iptg* inducible promoters, studied in *escherichia coli* and *pseudomonas fluorescens*. *Curr. Microbiol.*, 36(6):341–347.
- Jacob, F. and Monod, J. (1961). Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol.*, 3(3):318–356.
- Kandhavelu, M., Mannerstrom, H., Gupta, A., Hakkinen, A., Lloyd-Price, J., Yli-Harja, O., and Ribeiro, A. (2011). In vivo kinetics of transcription initiation of the *lar* promoter in *escherichia coli*. evidence for a sequential mechanism with two rate-limiting steps. *BMC Sys. Biol.*, 5:149.
- Lutz, R., Lozinski, T., Ellinger, T., and Bujard, H. (2001). Dissecting the functional program of *escherichia coli* promoters: the combined mode of action of *lac* repressor and *arac* activator. *Nuc. Acid. Res.*, 29(18):3873–3881.
- Makela, J., Kandhavelu, M., Oliveira, S., Chandraseelan, J. G., Lloyd-Price, J., Peltonen, J., Yli-Harja, O., and Ribeiro, A. (2013). In vivo single-molecule kinetics of activation and subsequent activity of the arabinose promoter. *Nucl. Acids Res.*, in press.
- McClure (1985). Kinetics of open complex formation between *Escherichia coli* rna polymerase and the *lac uv5* promoter. evidence for a sequential mechanism involving three steps. *Biochem.*, 24(11):2712–2723.
- Megerle, J., Fritz, G., Gerland, U., Jung, K., and Radler, J. (2008). Timing and dynamics of single cell gene expression in the arabinose utilization system. *Biophys. J.*, 95:2103–2115.
- Miller, O., Hamkalo, B., and Thomas, C. (1970). Visualization of bacterial genes in action. *Science*, 169(3943):392–395.
- Ribeiro, A. (2008). Dynamics and evolution of stochastic bistable gene networks with sensing in fluctuating environment. *Phys. Rev. E*, 78(1):061902.
- Ribeiro, A. (2010). Stochastic and delayed stochastic models of gene expression and regulation. *Math. Biosci.*, 233:1–11.
- Ribeiro, A. and Kauffman, S. (2007). Noisy attractors and ergodic sets in models of gene regulatory networks. *J. Theor. Biol.*, 247:743–755.
- Ribeiro, A. and Lloyd-Price, J. (2007). Sgn sim, a stochastic genetic networks simulator. *Bioinf.*, 23(6):777–779.
- Ribeiro, A., Zhu, R., and Kauffman, S. (2006). A general modeling strategy for gene regulatory networks with stochastic dynamics. *J. Comp. Biol.*, 13(9):1630–1639.
- Roussel, M. and Zhu, R. (2006). Validation of an algorithm for delay stochastic simulation of transcription and translation in prokaryotic gene expression. *Phys. Biol.*, 3:274–284.
- Sajbidor, J. (1997). Effect of some environmental factors on the content and composition of microbial membrane lipids. *Crit. Rev. Biotechnol.*, 17(2):87–103.
- Samoilov, M., Arkin, A., and Ross, J. (2002). Signal processing by simple chemical systems. *J. Phys. Chem. A*, 106:10205–10221.
- Schnappinger, D. and Hillen, W. (1996). Tetracyclines: antibiotic action, uptake, and resistance mechanisms. *Arch. Microbiol.*, 165(6):359–369.
- Talwalkar, A. and Kailasapathy, K. (2003). Metabolic and biochemical responses of probiotic bacteria to oxygen. *J. Dairy Sci.*, 86(8):2537–2546.
- Taniguchi, Y., Choi, P., Li, G., Chen, H., Babu, M., Hearn, J., Emili, A., and Xie, X. (2010). Quantifying *e. coli* proteome and transcriptome with single-molecule sensitivity in single cells. *Science*, 329(5991):533–538.
- Yamamoto, K. and Ishihama, A. (2005). Transcriptional response of *escherichia coli* to external zinc. *J. of Bacteriol.*, 187(18):6333–6340.
- Yu, J., Xiao, J., Lao, K., and Xie, X. (2006). Probing gene expression in live cells, one protein molecule at a time. *Science*, 311:1600–1603.
- Zhu, R., Ribeiro, A. S., Salahub, D., and Kauffman, S. (2007). Studying genetic regulatory networks at the molecular level: Delayed reaction stochastic models. *J. Theor. Biol.*, 246(4):725–745.

# Crawling Posture Learning in Humanoid Robots using a Natural-Actor-Critic CPG Architecture

Cai Li, Robert Lowe and Tom Ziemke

Interaction Lab  
University of Skovde  
Skovde, Sweden

Email: gauss.lee, robert.lowe, tom.ziemke@his.se

## Abstract

In this article, a four-cell CPG network, exploiting sensory feedback, is proposed in order to emulate infant crawling gaits when utilized on the NAO robot. Based on the crawling model, the positive episodic natural-actor-critic architecture is applied to learn a proper posture of crawling on a simulated NAO. By transferring the learned results to the physical NAO, the transferability from simulation to physical world is discussed. Finally, a discussion pertaining to locomotion learning based on dynamic system theory is given in the conclusion.

## Introduction

Crawling, as a type of quadrupedal locomotion, has been investigated on a number of humanoid robot platforms (Degallier et al., 2008) (Li et al., 2011). Compared to bipedal walking, crawling may be more versatile in terms of posture. For example, human beings can actually do *standard crawling* (knees and hands), *bear crawling* (feet and hands), *crab crawling* (upside down bear crawling), *leopard crawling* (a military specific crawling with elbows) (Wikipedia, 2013). Moreover, each of the crawling types differs regarding posture and needs specific training to perform. In this article, the work presented will focus on modelling *standard crawling* and training the NAO robot to learn the optimal posture.

From a bio-inspired perspective, central pattern generators (CPGs) are widely used in modelling locomotion on different robotic platforms (Ijspeert, 2008) (Harischandra et al., 2011) (Zhao et al., 2012), including humanoids (Degallier et al., 2008) (Endo et al., 2008). However, none of the above-cited work involves the posture optimization/learning for crawling and yet the posture determines the type of crawling. As far as CPG architectures are concerned, posture adjustment also plays an important role in the adaptability of CPGs (Grillner et al., 2008) (Orlovsky et al., 1999). Grillner et al (Grillner et al., 2005) not only posit that sensory feedback and postural control interactively connect the perceived environmental change to human neural structures via CPGs but also point out brainstem and basal ganglia are the

two main brain-related parts implicated in the adaptive solutions for locomotion. The latter theoretically implies the role of reinforcement learning (RL) as one possible implementable solution for presenting locomotion capabilities on humanoids.

RL provides an agent the capabilities of learning based on the interaction of the body, the environment and the neural structure. RL is one type of affective-modulation mechanism based on searching solutions for maximizing the reward-related value function (Sutton and Barto, 1998). Grillner et al's (Grillner et al., 2008) brain-based perspective on locomotion implies the role of RL in the adaptation of CPGs. Biologically, some scientists assume it accounts for the functions of basal ganglia in human brains (Wiering and van Otterlo, 2012). This neural-anatomic link hints that RL might also be implicated in locomotion and specifically, learning to crawl. Recently, the emerging new methods of continuous-space learning links machine learning closely to RL, such as continuous action space learning automaton (Cacla) and natural actor critic (NAC) (van Hasselt and Wiering, 2007) (Peters and Schaal, 2008). Both of these methods are used in different motor learning tasks on robots to update in a relatively high-dimensional parameter space (Kober et al., 2012) (Farkas et al., 2012). In our work, because of relatively high learning efficiency of NAC, it is used to figure out the optimal posture of crawling under the guidance of a specific reward function. RL, as an approach independent from traditional supervised and unsupervised learning, is a method which can seamlessly integrate scaffolding-related (supervised) and self-learning factors (unsupervised) into one process.

In this article, a posture learning architecture on a crawling humanoid is presented. In section 2, the main principles/theories pertaining to emulating infant crawling on a humanoid robot are introduced in detail, including the theories of CPGs and RL. In section 3, with the results from the simulation, the learned models are transferred to the physical robot for verification. The statistical results are given and analyzed. In section 4, a conclusion related to RL and learning locomotion is drawn.



## Methods and Theories

### CPGs and Crawling

Since crawling is a periodic motion, using CPGs to interpret and model it falls within the scope of the biological explanation of locomotion (Grillner et al., 2008) (Scott L Hooper, 2001). Via transferring this knowledge to robotics, it provides new methodologies for understanding morphology constrained locomotion principles on different robots (Ijspeert, 2008), including crawling robots (Harischandra et al., 2011) (Zhao et al., 2012) (Degallier et al., 2008). The approaches for modelling crawling can be classified into two categories: the engineering and the bio-inspired. Specifying the pre-defined trajectories of end-effectors, Aoi et al (Aoi and Tsuchiya, 2005) propose an engineering-based crawling model implemented on a bipedal robot. A key drawback of this approach is that this crawling robot might need recalculation and even remodeling when the environment changes, which limits motoric adaptation capabilities. On the basis of this, a lot of bio-inspired roboticists posit other solutions based on CPGs for avoiding the limitations of engineering methods. Nakamura et al, Righetti et al and Li et al (Nakamura et al., 2007) (Righetti and Ijspeert, 2008) (Li et al., 2013) together applied CPGs to locomotion by regarding sensory feedback as the input to reshape the output of neural controllers, endowing their architectures with *self-adaptation capabilities*. In this article, the CPG architecture based on sensory feedback reshaping is coined as *least sensory feedback CPG* model. The definition is as follows:

**Least Sensory Feedback CPG Model:** The adaptation and feasibility of CPG models reply on a limited/necessary number of sensors so that the output of the model can be interactively reshaped by perceived contextual change. The CPG architecture with above-mentioned characteristics is defined as the least sensory feedback CPG.

“least” here emphasizes necessary/minimal. For example, based on dynamic systems theory, Righetti et al propose the four-cell CPG network with necessary fast/slow transition feedback to implement standard crawling on the iCub robot (Degallier et al., 2008). Without fast/slow transition, the iCub cannot crawl in the real world. Inspired by Righetti et al’s model, we adopt the fast/slow transition into our experiments as a necessary self-modulation mechanism of CPGs.

**the mathematical model of CPGs** Inspired by work on iCub crawling (Degallier et al., 2008) and adapted from our previous work (Li et al., 2011), the Hopf oscillators are adapted into our work:

$$\dot{x}_i = a(m - x_i^2 + y_i^2)y - \omega_i x_i \quad (1)$$

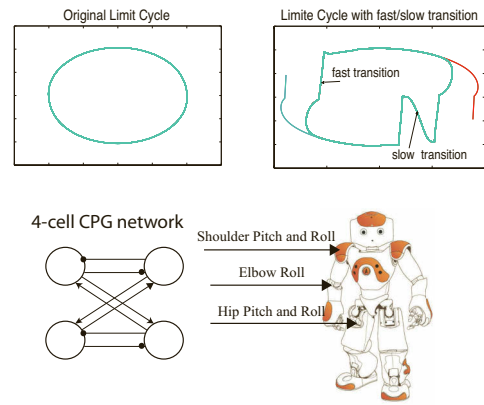


Figure 1: Top: the limit cycles with and without transition feedback (Right and Left). Bottom-left: the four-cell CPG network. The round-head and sharp-head arrows represent negative (-1) and positive (+1) connection weights respectively. Bottom-right: The NAO robot and its correspondent joints under control of the CPG network for crawling.

$$\dot{y}_i = b(m - x_i^2 + y_i^2)x + \omega_i y_i + \sum_j \alpha_{ji} y_{ij} + F_i \quad (2)$$

$$\omega_i = 2 \cdot \pi \left( \frac{\omega_{swing}}{(1 + e^{-100y_i})} + \frac{\omega_{stance}}{(1 + e^{100y_i})} \right) \quad (3)$$

where the  $x_i$  is the output of the Hopf Oscillator and  $y_i$  is the internal state.  $m$  is the amplitude and  $a, b$  are the convergence rate ( $m=1$ ,  $a=1$  and  $b=5$  for all the experiments).  $\omega_i$  is the internal weight in this coupled oscillator. It is usually set to 1.  $y_{ij}$  is the output of the other cells except cell  $i$  and  $\alpha_{ji}$  is the external weight (from cell  $j$  to  $i$ ) of the four-cell network (Figure 1:bottom-left). Meanwhile,  $\omega_i$  also represents the frequency of this oscillator. Interestingly, by changing values of  $\omega_{swing}$  and  $\omega_{stance}$ , you can change the duration of increase and decrease rate of the oscillator. According to the statistical verification of Righetti et al’s model (Degallier et al., 2008), it is best to set  $\omega_{swing} = \omega_{stance} = 1.0$ .  $F_i$  is the fast/slow transition feedback which can be regarded as time-reset afferent feedback in the first layer of our previous model (Li et al., 2013) and defined as follows:

$$\begin{aligned} & \text{if } y_i > 0, y_i < 0 \text{ and } x_i \geq 0 \\ & F_i = -\text{sign}(y_i) * 50 \end{aligned} \quad (4)$$

$$\begin{aligned} & \text{if } x_i < -0.45, y_i \geq 0 \text{ or } x_i > 0, y_i \leq 0 \\ & F_i = -2\omega_i y_i - \sum_j \alpha_{ji} y_{ij} \end{aligned} \quad (5)$$

Where equation (4) describes the fast transition in which  $y_i$  decreases very quickly and equation (5) represents the slow transition in which the increase of  $y_i$  is delayed. Figure 1 (top) shows the comparison of limit cycles before and after adding fast/slow transition. The fast transition aims to control the robot to swing back quickly and the slow transition,

based on a triggered delay in the oscillator, is to smoothen the transition from stance phase to swing phase. These two characteristic types of feedback are activated by pressure sensors in the crawling iCub but maintained as invariant features in our work as the NAO robot does not have usable pressure sensors. They are regarded as types of proprioceptive sensory feedback related to the position of the joint. Then the investigation of crawling in NAO is transformed to the problem of how to find a proper posture that allows the robot to crawl with a crawling-featured “limit cycle”.

**Standard crawling and its key variables** Crawling, as the first milestone motion ability in human infants (Clearfield, 2004), serves as a cornerstone gateway to learn the body-environment interaction (Kail and Cavanaugh, 2012). Most expert crawlers during infancy move on knees and hands according to Righetti et al.’s investigation (Righetti et al., 2008). Hereby, in order to emulate standard crawling on a robot, the knowledge of how infants learn to crawl might be applicable. Clearfield et al (Clearfield, 2004) emphasize the salient role of spatial-temporal memory, especially the “distance”, in the process of infant crawling. Adolph et al (Adolph et al., 2012) found that the parental scaffolding by holding up crawling infants offers positive “safety” assurance on the posture. Therefore, it might be necessary to immerse two factors (distance and posture) into the learning mechanism. The posture can be embodied as the angle affecting the head direction (Figure 2). In our work, it is controlled by a Gaussian distribution between 30 degree and 50 degree.

The number of joints for a humanoid pertaining to the whole-body posture can be very high dimensional. To avoid high dimensionality, firstly, the robot begins in a crawling posture with which it stably mounts on knees and hands. Then only the necessary joints are taken into account for adjusting the crawling posture, including shoulders (pitch and roll), hips (pitch and roll) and elbows (roll). The knee and ankle joints are neglected as they do not move so much for crawling (Righetti et al., 2008).

### Natural Actor Critic in Posture Learning

**Episodic Natural Actor Critic (eNAC)** eNAC is well-known for its learning efficiency on searching optima in a continuous parameterized space. As the posture adjustment requires a continuous adaptation, eNAC is selected to optimize the crawling posture. Compared to Cacla, another efficient continuous-space RL, eNAC might suffer in possible failures by updating a parameterized model into an unknown action space (Wiering and van Otterlo, 2012). So a positive eNAC is applied in our work (for details see part 2) to highlight the potential failures.

NAC was proposed by Kakade et al (Kakade, 2001) and further developed and used in motor learning by Peters et al (Peters, 2007) (Peters and Schaal, 2008). It transforms the

traditional RL problem on solving the Bellman equation to an explorative process of linear regression. As a policy gradient approach, the principles of NAC are elucidated as follows:

Assume the stationary policy is  $\pi^\theta(\mathbf{x}, \mathbf{u})$  which can determine action space  $\mathbf{u}$  based on state space  $\mathbf{x}$  with a static distribution  $d^\pi(x)$ . The immediate reward is  $r(x, u)$ . Then the expected reward  $J(\theta)$  can be written as:

$$J(\theta) = \int_{\mathbf{x}} d^\pi(\mathbf{x}) \int_{\mathbf{u}} \pi^\theta(\mathbf{u}|\mathbf{x}) r(\mathbf{x}, \mathbf{u}) d\mathbf{x} d\mathbf{u} \quad (6)$$

$$\theta_{n+1} = \theta_n + \alpha \nabla_\theta J|_{\theta=\theta_n} \quad (7)$$

where the policy  $\pi^\theta(\mathbf{x}, \mathbf{u})$  is derivable at the policy parameters  $\theta$ , namely  $\nabla_\theta \pi^\theta$  exists. For maximizing expected reward  $J(\theta)$  with respect to  $\theta$ , the policy gradient will find the steepest increase direction  $\nabla_\theta J = J(\theta + \Delta\theta) - J(\theta)$  to update searching policy  $\pi^\theta(\mathbf{x}, \mathbf{u})$  until it converges.  $n$  represents the  $n$ th step of the update and  $\alpha$  is the learning rate (equal to 0.01). By and large, Equation (6) and (7) plot the rudimentary rule of thumb for policy gradient approaches. Transformed to natural policy gradient, Equation (6) and (7) can be revised to (8) and (9):

$$\nabla_\theta J(\theta) = \int_{\mathbf{x}} d^\pi(\mathbf{x}) \int_{\mathbf{u}} \pi^\theta(\mathbf{u}|\mathbf{x}) \nabla_\theta \log(\pi^\theta(\mathbf{u}|\mathbf{x})) \nabla_\theta \log^T(\pi^\theta(\mathbf{u}|\mathbf{x})) \mathbf{w} d\mathbf{x} d\mathbf{u} = F_\theta \mathbf{w} \quad (8)$$

$$\theta_{n+1} = \theta_n + \alpha F_\theta^{-1} \nabla_\theta J|_{\theta=\theta_n} \quad (9)$$

$$F_\theta = \int_T \pi^\theta \nabla_\theta \log \pi^\theta \nabla_\theta \log \pi^\theta d\theta$$

where  $F$  is the Fisher Matrix (FM). Multiplied by FM, normal policy gradient is changed to the steepest one (Here all the  $\mathbf{x}, \mathbf{u}$  are neglected for simplification reason).  $\mathbf{w}$  is a weight vector of the linear approximation and  $\nabla_\theta \log^T(\pi^\theta(\mathbf{u}|\mathbf{x}))$  is the group of basis functions. Then conclusively, by replacing  $\nabla_\theta J(\theta)$  in (9) with (8), the natural PG becomes:

$$\theta_{n+1} = \theta_n + \alpha \mathbf{w} \quad (10)$$

The RL problem changes from searching the steepest policy gradient to a normal regression problem with basis functions. Since the state-action function  $Q^\pi(\mathbf{x}, \mathbf{u}) = b(\mathbf{x}) + \nabla \log(\pi^\theta(\mathbf{u}|\mathbf{x})) \mathbf{w}$  and  $Q^\pi(\mathbf{x}, \mathbf{u}) = r(\mathbf{x}, \mathbf{u}) + \lambda \int_{\mathbf{x}'} p(\mathbf{x}'|\mathbf{x}, \mathbf{u}) V(\mathbf{x}') d\mathbf{x}'$  (where  $\lambda$  is the discounting factor,  $\mathbf{x}'$  is the next state,  $p(\mathbf{x}'|\mathbf{x}, \mathbf{u})$  is the probability of state transition.), assume the value function is  $V(\mathbf{x}) = b(\mathbf{x})$  and can be approximated by  $\psi^T(\mathbf{x}) \mathbf{v}$  (where  $\mathbf{v}$  is the weight vector and  $\psi$  is the vector of basis function related to the value function.) (Baird, 1993). Therefore, the approximation can be re-written:

$$\nabla \log^T(\pi^\theta(\mathbf{u}_t|\mathbf{x}_t)) \mathbf{w} + \psi^T(\mathbf{x}_t) \mathbf{v} = r(\mathbf{x}_t, \mathbf{u}_t) + \lambda \psi^T(\mathbf{x}_{t+1}) \mathbf{v} + \epsilon(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{u}_t) \quad (11)$$

if we sum up left and right side of equation (11), we get a straightforward regression problem for H episodes with s time-length:

$$\left(\sum_{t=1}^s \alpha_t \nabla \log^T(\pi^\theta(\mathbf{u}_t|\mathbf{x}_t))\right)_{1:H} \mathbf{w} + J = \left(\sum_{t=1}^s \alpha_t r(\mathbf{x}_t, \mathbf{u}_t)\right)_{1:H}$$

where  $J$  is the value-function related term considered as a constant baseline and  $\alpha_t$  is the average discounting factor. By means of the least square learning rule, the natural PG  $\mathbf{w}$  can be obtained for H episodes:

$$\begin{bmatrix} \mathbf{w} \\ J \end{bmatrix} = (\phi \phi^T)^{-1} \phi R.$$

$$\phi = \left[\sum_{t=1}^s \alpha_t \nabla \log^T(\pi^\theta(\mathbf{u}_t|\mathbf{x}_t)), 1\right]_{1:H}^T \quad (12)$$

$$R = \left[\sum_{t=1}^T \alpha_t r(\mathbf{x}_t, \mathbf{u}_t)\right]_{1:H}^T \quad (13)$$

where  $1 : H$  represent H times samplings within one trial (refer to details in the Algorithm).

**Implementation of the learning algorithm** Based on the above-mentioned proof (for details, please refer to (Peters, 2007)), the state space and action space are defined as  $\mathbf{P} \sim [P_{shoulder}, P_{hip}, P_{elbow}]$  and  $\mathbf{U} \sim [U_{shoulder}, U_{hip}, U_{elbow}]$ . So  $P = P_0 + U$ , where  $P_0$  is the initial posture vector. With the parametrized policy  $\pi^\theta(\mathbf{U}|\mathbf{P})$ , a posture can be learned via the optimization of the gaussian policy:

$$\begin{aligned} \pi_\theta(\mathbf{U}, \mathbf{P}) &= N(\mathbf{U}, \bar{\mathbf{U}}, \sigma) \\ &= \frac{2\pi}{\sigma} \exp\left(-\frac{(\mathbf{U} - \bar{\mathbf{U}})(\mathbf{U} - \bar{\mathbf{U}})^T}{\sigma^2}\right) \end{aligned}$$

where  $\mathbf{U}$  is the output vector of the policy and  $\bar{\mathbf{U}}$  is the input vector equal to the updated  $\mathbf{U}$  from last trial and  $\bar{\mathbf{U}} \sim [\theta_{shoulder}, \theta_{hip}, \theta_{elbow}]$ .  $\sigma$  is the exploration rate which determines the variance of  $\mathbf{U}$  from  $\bar{\mathbf{U}}$ . In our work,  $\sigma = 0.03$  as if  $\sigma > 0.05$ , the posture updating is unstable and if  $\sigma < 0.01$ , the posture adjustment is too insensitive and time-consuming.

The schema of eNAC algorithm is shown below:

**eNAC Algorithm:**

Repeat M trials each of which includes 10 rollouts (H=10), use the policy  $U_{1:H} \sim \pi_\theta(\mathbf{U}, \mathbf{P})$  to generate H groups of actions, each action is taken for time t=1,2,...s.

Calculate:

for each rollout, the episodic return  $r_i =$

$(\sum_{t=1}^s \alpha_t r(\mathbf{x}_t, \mathbf{u}_t))$   
the eligibility  $\psi_i = [\nabla \log^T(\pi^\theta(\mathbf{u}_t|\mathbf{x}_t)), 1]_i$   
then the gradient is:

$$\begin{bmatrix} \mathbf{w} \\ J \end{bmatrix} = (\phi \phi^T)^{-1} \phi R.$$

where  $R = [r_1, r_2, \dots, r_H]^T$  and  $\phi = [\psi_1, \psi_2, \dots, \psi_H]^T$   
Updating for each trial:

if  $\delta > 0$ , with  $\delta = R_{avg} - V_n$  where  $R_{avg}$  is the average of R and  $V_n$  is the episodic value function of last updating:

$V_{n+1} = V_n + 0.1 * \delta$  and  $\theta_{n+1} = \theta_n + \alpha \mathbf{w}$ , otherwise no updating.

Until the convergence condition is satisfied:  $\delta < 0$  all the time or  $\delta < |10^{-4}|$ .

It is noteworthy that, inspired by the Cacla architecture (van Hasselt and Wiering, 2007), the ‘‘positive updating’’ is used to avoid the inappropriate updating in the parameterized action space. Since the function approximation cannot accurately converge to the real Q function ( $\mathbf{w}$  cannot be zero), the convergence condition is necessary to determine the termination of each learning process.

## Experimental Setup

As mentioned in Section 1, the crawling distance and spine-line angle are the two important variables for evaluating the crawling posture (Figure 2). Accordingly, in RL, the reward function is composed of two terms ( $r_{distance}$  and  $r_{angle}$ ) related to the two factors that represent the two above-mentioned variables:

$$r_{reward} = r_{distance} + r_{angle}$$

$$r_{distance} = \exp\left(\frac{D}{2}\right) - 1$$

$$r_{angle} = \exp(e) - 1$$

$$\text{with } e = N(\mathbf{x}_0 = 1.05, \sigma = 0.1)$$

where D is the distance the robot crawls every episode. e is a Gaussian distribution with the center 1.05 (approximately 45 degree) and variance 0.1. As a matter of fact, distance is a very important measure of the improvement of crawling and spine-line angle is used to control and stabilize the body-height alteration.

In the experiments described below, the crawling CPGs are firstly implemented on a simulated NAO robot in Webots which is a popular commercialized robot simulator (Michel, 2004). Then the statistically-learned models are transferred to the physical robot for testing. The advantage of using the simulated robot is not only to avoid unexpected damage upon the physical robot but also to simplify the measurement of distance by using the special supervisor functions of Webots.

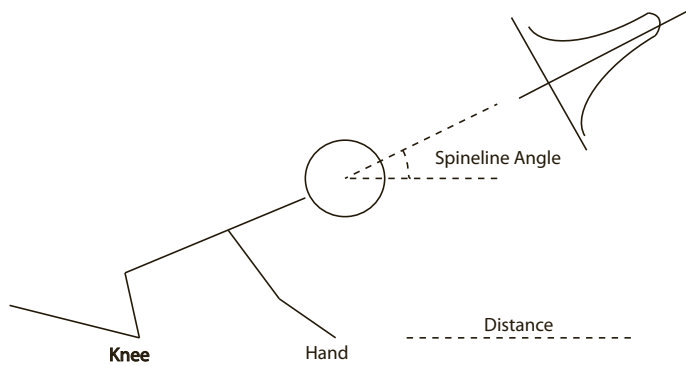


Figure 2: The standard crawling posture on knees and hands. The distance and spinline angle indicate the quality of crawling. The spinline angle is limited in a scope (30 degree~60 degree) by a Gaussian function.

### Experiment 1: Utilizing proprioceptive sensory feedback

This experiment is to verify the importance of proprioceptive sensory feedback (PSF) characterizing the crawling CPG. The snapshots of our crawling video (Figure 3) shows the utility of PSF. In this experiment, the NAO robot crawls with a manually-tuned posture. From the video(Li, 2013b), it is clearly observed that the NAO robot can crawl much further/faster with PSF. The multiple runs do not qualitatively effect the performance of crawling with possible simulated noise effects.

### Experiment 2: Learning the posture of standard crawling

With the crawling CPG and its proprioceptive sensory feedback, eNAC is applied on the simulated NAO robot for learning the optimal posture. Each episode of eNAC lasts 25s and the posture is continuously optimized by maximizing the crawling distance (in the same time duration) within the scope of the spinline angle. Each run of the experiment takes approximately 7 hours to converge (on a single Mac-Book pro). In total, 10 results are obtained statistically by running the experiment with variable values reset at the beginning of each run 10 times. Figure 4 shows the snapshots of optimized crawling and its comparison to the original one. Based on the video(Li, 2013c) and the snapshots, we can summarize the observable difference of two crawling gaits before/after learning. Firstly, before learning, the robot struggles to crawl forward with an obvious slippage on hands but after learning, the slippage disappears. Secondly, the crawling distance has been well optimized after learning. Subsequently, a detailed understanding of how the posture is adjusted by the eNAC is explained in the next section.

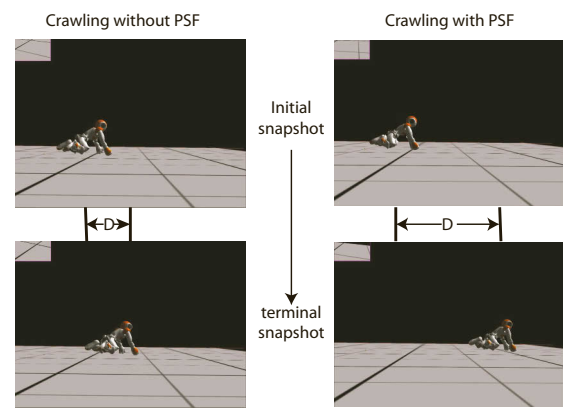


Figure 3: Left: the performance comparison of crawling without proprioceptive sensory feedback (PSF) between the initial and terminal snapshot (from top to bottom). Right: the performance comparison of crawling with PSF. D represents the rough measurement of the distance the robot crawls in the video. Both videos are shot from the same angle within the same time period.

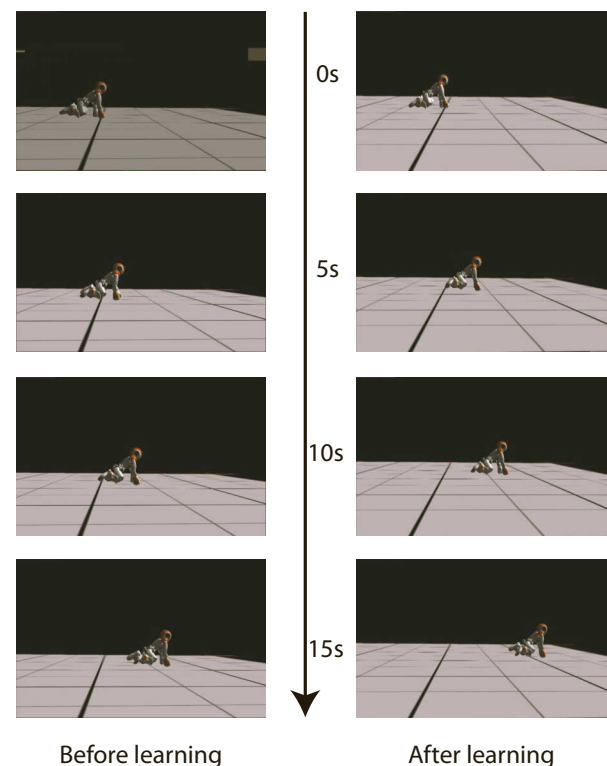


Figure 4: Left: the performance of crawling before learning. Right: the performance of crawling after learning. The left/right column snapshots are casted at the same time 0s, 5s ,10s, 15s and compared correspondently. The distance the robots crawls is labeled by three feature lines: one bold line (starting line) and two solid lines (distance mark lines).



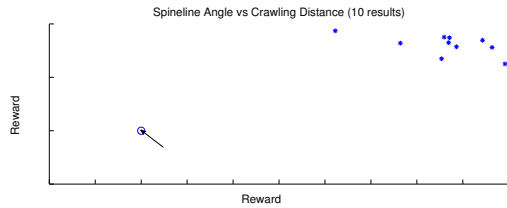


Figure 5: The learning results of spine angle vs distance. The circle shows the values of  $r_{angle}$  and  $r_{distance}$  at the initial state. The cross points are the results after learning.

## Results and Analysis

### Results analysis

Since every run of each experiment starts independently from the same initial posture, it is intriguing to see how the statistical learning process tunes the six joints together and also if the 10 results are consistent. The typical characteristic of eNAC is that it can find the correct updating directions in the parameter space with a relatively high learning efficiency. Even though it might suffer in leading the updating into uncertain/invalid directions (Wiering and van Otterlo, 2012), the positive eNAC used in this article could possibly avoid this drawback. Figure 7.A~B show the increase of reward during the 10-run experiment. All 10 runs show the reward levels are boosted based on postural and distance rewards. Moreover, in almost every run of the experiment, the reward is effectively augmented during the first 20 episodes. This implies that positive eNAC could seek out valid updating pathways in a short time.

From the other point of view, the details of joint-value updating are highlighted in the Figure 7.C~H. Generally, even though different runs converge to distinct results (marked by black circles) within disparate updating episodes, the updating curves of each joint in every run share qualitatively similar update profiles and converge towards same directions. Therefore, above-mentioned conclusive summary insinuates the consistency over independent runs of learning processes. As a statistical learning approach, all the results obtained in the 10-run experiment are not the same. However, the converged posture adjustment values are located in a close scope. For example,  $\bar{\mathbf{U}} \sim [\theta_{HipYawPitch}, \theta_{HipRoll}, \theta_{HipPitch}, \theta_{ShoulderPitch}, \theta_{ShoulderRoll}, \theta_{ElbowRoll}]$  converges to the values approximately around 0.05 rad, 0.1 rad, -0.15 rad, -0.1 rad, -0.15 rad, 0.1 rad respectively (observed from Figure 7.C~H).

Figure 5 displays the improvement of crawling gait in two aspects. Compared to the initial state, the crawling distance and spine angle are both improved. In this article, the standard crawling posture is maintained by  $r_{angle}$  which defines the scope of spine angle. It somewhat stabilizes the crawling posture adjustment. In the experiment, it is observed that if the angular feedback mechanism is

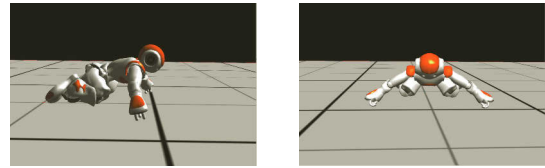


Figure 6: The scorpion-like posture unexpectedly learned without the control of the spine angle.

omitted, the robot produces a crawling behaviour (flat scorpion-like motion) that does not conform to any typical infant crawling gaits (Figure 6). Although the scorpion-like crawling is an optimized solution in terms of only the distance, it is still considered to be an improper result as non-standard crawling posture. The crawling distance (in the same time duration) is a measure indicating the embodied improvement of crawling gait. The two different rewards are two reinforced drivers to make a proper crawling gait emerge.

### Transferrability to the physical robot

The 10 results obtained via eNAC are transferred to test on the physical robot. The transferability from the simulated robot to the physical robot is always a significant problem. The main reasons causing the failure of transferring are: firstly, the inappropriate physics engine might lead to distinct physical interaction in the simulator compared to the real world, including the collision detection which is very important for locomotion modelling. Webots uses a widely-used physics engine, ODE (Open Dynamics Engine (Michel, 2004)). It has attained reliable performance in robotic simulation, especially on locomotion-related tasks (Harischandra et al., 2011) (Degallier et al., 2008). Secondly, the difference in timing. The machine time might be different from the real time. In our work, the physical robot cannot crawl with the straightforward transferring from the simulated results. However, the crawling gait is successfully transferred by doubling the CPG frequency. It seems the numerical integration used to calculate CPG has a strong attachment to the speed of the machine, causing a difference in the CPG timing on different hardware. Therefore, in order to transfer the work in the simulator to the physical robot, the timing relation between the simulation computer and robot's hardware has to be identified.

Figure 8 shows the gait performance on the physical robot with one crawling step (for the video, refer to (Li, 2013a)). Authors tested 10 results of the learning. 7 out of 10 can successfully perform a smooth crawling gait. The other 3 failed as the low value of  $\theta_{ShoulderPitch}$  cannot allow the arm to lift above the ground completely when the two arms are alternating. The robot cannot crawl "smoothly" even though it can crawl nicely in the simulator. This might be caused by the difference between the simulated world based on the

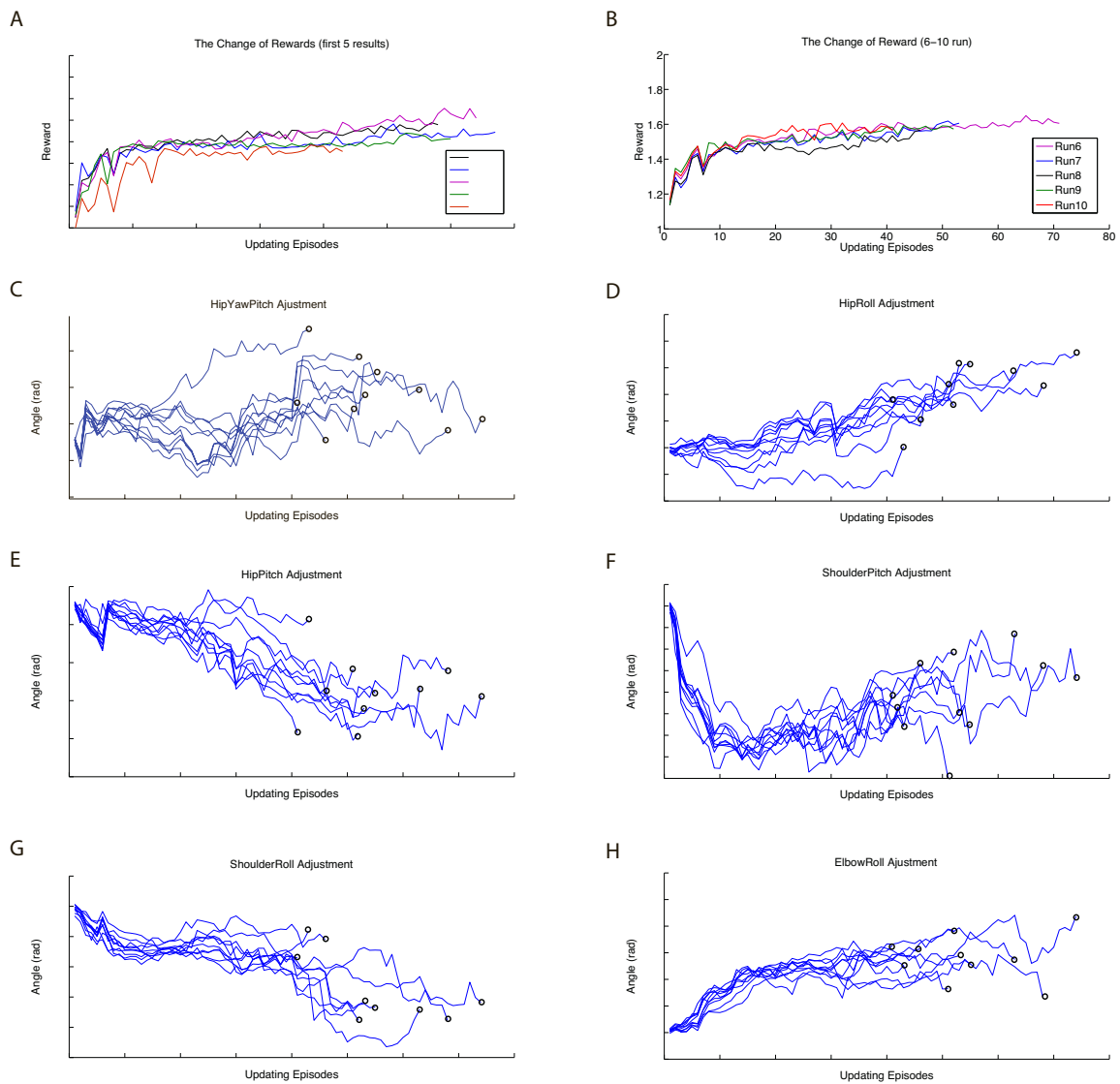


Figure 7: Figure 7.A~B show the reward increase of 10-run experiments. The "updating episodes" the episode which satisfies the updating condition of eNAC. Figure 7.C~H depict the change of joint values during the learning process and it covers the results of 10 runs for each joint. Those joints include HipYawPitch, HipRoll, HipPitch, ShoulderPitch, ShoulderRoll and ElbowRoll, the six joints affecting the quality of crawling gait. The black circles mark the converged points of each joint.

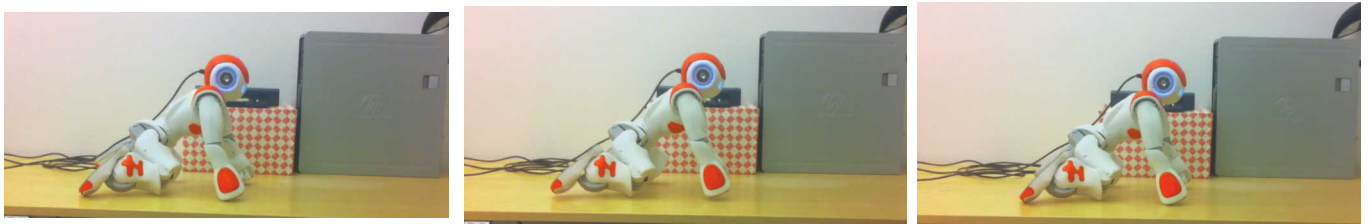


Figure 8: The implementation on the physical robot. This figure shows the video snapshot of one-crawling-step NAO robot on a wooden flat table (One crawling step means one time alternation of the supporting leg and arm.).

physics engine in Webots and the real physical world.

## Conclusion

In the previous two sections, we verified the necessity of proprioceptive sensory feedback and optimized postures for the crawling gait and tested the gait on the physical robot successfully. The results of experiments highlight the importance of sensory feedback and postural control for the CPG architecture. With a complete least-sensory-feedback CPG, the sensory feedback is designed to reshape the CPG output and the posture control is involved in shifting the oscillation centers of CPGs. Grillner et al.' work also implies these two functions in the CPG architecture(Orlovsky et al., 1999)(Grillner et al., 2008). Using eNAC, it assists the robot to learn/adapt on the ground of the interaction with the environment.

## Conclusion

### From Grillner's Inspiration towards Reality

Grillner et al's work on the theories of CPGs might be one of the most complete collections for CPG "huggers". His group not only investigate the biological theories of CPGs spanning from mollusc to man(Orlovsky et al., 1999) but also propose the plausible architecture of a goal-directed locomotion system which links CPGs to brain in an exact way(Grillner et al., 2005)(Grillner et al., 2008). This connection between the brain and CPGs is read as the connection between RL and oscillators in this article. RL fulfills the functions of basal ganglia and MLR (mesencephalic locomotor region) highlighted by Grillner et al(Grillner et al., 2008). The roles of CPGs include sensory-feedback integration and posture control. With the least-sensory-feedback concept, in this article, authors simply interpret roles of CPGs as the output reshaping and postural learning. Even though the crawling posture is successfully learned in our work, several questions still remain unsolved: Firstly, the least-sensory-feedback CPG model might not be the best solution to explaining sensory feedback integration. With pre-defined sensory feedback, even though the CPG model can acquire the adaptation in a lot of work(Li et al., 2011)(Li et al., 2012)(Li et al., 2013)(Nassour et al., 2013)(Nakamura et al., 2007), it is still functionally limited. If the sensor configuration is altered or the robot lacks necessary sensors (the case in this article), the CPG architecture might lose its adaptation to the changing environment. Therefore, a general solution for integrating sensory feedback may be demanded. Secondly, the energy efficiency of CPGs. Energy efficiency is a highlighted property of using CPGs(Tomoyuki et al., 2009)(Matsuo et al., 2012). Even though CPGs work as trajectory generators in our work, the energy efficiency is actually, at least, not high as the stiffness of joints is maintained as constant 0.9 (0~1.0 for the NAO robot). The link between energy efficiency and CPGs seems not be clear in theories. It might be a factor also related to morphologies. Since the

NAO robot has a rigid body, it might be difficult to achieve high energy efficiency.

## Rethinking locomotion learning

The focus of locomotion modelling has been gradually expanded into bio-inspired approaches(Ijspeert, 2008). Pfeifer et al emphasize the role of body morphologies as a core value of soft robotics(Pfeifer and Bongard, 2006). The locomotion learning/modelling has been elucidated as an interactive process instead of only being based on engineering approaches. The interaction of neural controllers, the body and the environment is the core of this ideology. From this angle, RL, as an interactive learning process, is able to bridge the three parts. Especially, with the newly proposed RL approached in continuous space, it can serve locomotion learning for more complicated parameterized models. This not only fits within the perspective of the architecture of Grillner et al's proposition(Grillner et al., 2008) but also is consistent with the theories of dynamic systems theory(Thelen, 1996).

## Drawbacks and Future work

Aside from the two major drawbacks mentioned above, the transferability from simulation to physical world is also an unresolved problem. In order to avoid this problem, the learning process should be able to transfer to the physical robot. In our work, the learning approach relies on the measurement of crawling distance. Webots proffers a convenient solution on any distance measuring with supervisors. In the physical world, the robot should have the ability to measure distance visually so that the learning process can be transferred. In future work, the motor primitive models(Kober et al., 2012) might be used to form a general solution to reshaping the CPG output. With this approach, the least-sensory-feedback model can possibly be adapted into a generic CPG model.

## Acknowledgment

The authors would like to thank the European Project Robot-DoC (<http://www.robotdoc.org>) for funding and providing help towards our work.

## References

- Adolph, K. E., Cole, W. G., Komati, M., Garciguire, J. S., Badaly, D., Lingeman, J. M., Chan, G. L. Y., and Sotsky, R. B. (2012). How do you learn to walk? Thousands of steps and dozens of falls per day. *Psychological science*, 23(11):1387–94.
- Aoi, S. and Tsuchiya, K. (2005). Transition from quadrupedal to bipedal locomotion. *Intelligent Robots and Systems*, pages 1998–2003.
- Baird, L. (1993). Advantage updating. *Neural Networks*, (6):2448–2453.
- Clearfield, M. W. (2004). The role of crawling and walking experience in infant spatial memory. *Journal of experimental child psychology*, 89(3):214–41.

- Degallier, S., Righetti, L., Natale, L., Nori, F., Metta, G., and Ijspeert, A. (2008). A modular bio-inspired architecture for movement generation for the infant-like robot iCub. *2008 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 795–800.
- Endo, G., Morimoto, J., Matsubara, T., Nakanishi, J., and Cheng, G. (2008). Learning CPG-based Biped Locomotion with a Policy Gradient Method: Application to a Humanoid Robot. *The International Journal of Robotics Research*, 27(2):213–228.
- Farkaš, I., Malík, T., and Rebrová, K. (2012). Grounding the Meanings in Sensorimotor Behavior using Reinforcement Learning. *Frontiers in neurorobotics*, 6(February):1.
- Grillner, S., Hellgren, J., Ménard, A., Saitoh, K., and Wikström, M. a. (2005). Mechanisms for selection of basic motor programs—roles for the striatum and pallidum. *Trends in neurosciences*, 28(7):364–70.
- Grillner, S., Wallén, P., Saitoh, K., Kozlov, A., and Robertson, B. (2008). Neural bases of goal-directed locomotion in vertebrates—an overview. *Brain research reviews*, 57(1):2–12.
- Harischandra, N., Knuesel, J., Kozlov, A., Bicanski, A., Cabelguen, J.-M., Ijspeert, A., and Ekeberg, O. (2011). Sensory feedback plays a significant role in generating walking gait and in gait transition in salamanders: a simulation study. *Frontiers in neurorobotics*, 5(November):3.
- Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots : a review Neurobiology of CPGs. *Neural Networks*, 21:642–653.
- Kail, R. V. and Cavanaugh, J. C. (2012). *Human Development: A Life-span View*. Cengage Learning.
- Kakade, S. (2001). A Natural policy gradient. *Advances in neural information processing systems*.
- Kober, J., Wilhelm, A., Oztog, E., and Peters, J. (2012). Reinforcement learning to adjust parametrized motor primitives to new situations. *Autonomous Robots*, 33(4):361–379.
- Li, C. (2013a). crawling before learning and after learning: <https://www.youtube.com/watch?v=5mygX30x96U>.
- Li, C. (2013b). Crawling: <http://www.youtube.com/watch?v=G65HgQPtsOY>.
- Li, C. (2013c). First crawling Nao with optimized posture: <https://www.youtube.com/watch?v=ABW60YK3Mbw>.
- Li, C., Lowe, R., Duran, B., and Ziemke, T. (2011). Humanoids that crawl: Comparing gait performance of iCub and NAO using a CPG architecture. *2011 IEEE International Conference on Computer Science and Automation Engineering*, pages 577–582.
- Li, C., Lowe, R., and Ziemke, T. (2012). Modelling Walking Behaviors Based on CPGs: A Simplified Bio-inspired Architecture. *From Animals to Animats 12 Lecture Notes in Computer Science*, 7426:156–166.
- Li, C., Lowe, R., and Ziemke, T. (2013). Humanoids Learning to Walk: a Natural CPG-Actor-Critic Architecture. *Frontiers in Neurorobotics*, 7.
- Matsuo, T., Sonoda, T., and Ishii, K. (2012). A Design Method of CPG Network Using Energy Efficiency to Control a Snake-Like Robot. In *2012 Fifth International Conference on Emerging Trends in Engineering and Technology*, pages 287–292. IEEE.
- Michel, O. (2004). Webots TM : Professional Mobile Robot Simulation. *Advanced Robotic Systems*, 1(1):40–43.
- Nakamura, Y., Mori, T., Sato, M.-a., and Ishii, S. (2007). Reinforcement learning for a biped robot based on a CPG-actor-critic method. *Neural networks*, 20(6):723–35.
- Nassour, J., Hugel, V., Ouezdou, F. B., Cheng, G., and Member, S. (2013). Failure Maps : Applied to Humanoid Robot Walking. *IEEE Transactions on Neural Networks*, 24(1):81–93.
- Orlovsky, G., Deliagina, T. G., and Grillner, S. (1999). *Neuronal Control of Locomotion From Mollusc to Man*. Oxford University Press.
- Peters, J. (2007). *Machine learning for motor skills in robotics*. PhD thesis.
- Peters, J. and Schaal, S. (2008). Natural Actor-Critic. *Neurocomputing*, 71(7-9):1180–1190.
- Pfeifer, R. and Bongard, J. C. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books)*. The MIT Press.
- Righetti, L. and Ijspeert, A. J. (2008). Pattern generators with sensory feedback for the control of quadruped locomotion. *2008 IEEE International Conference on Robotics and Automation*, pages 819–824.
- Righetti, L., Nylén, K. A., and Ijspeert, A. (2008). Is the locomotion of crawling human infants different from other quadruped mammals? Technical report.
- Scott L Hooper (2001). Central Pattern Generators. *Encyclopedia of Life Sciences*.
- Sutton, R. and Barto, A. (1998). Reinforcement Learning: An Introduction. *IEEE Transactions on Neural Networks*, 9(5):1054–1063.
- Thelen, E. S. (1996). *A dynamic systems approach to the development of cognition and action*. MIT Press.
- Tomoyuki, T., Azuma, Y., and Shibata, T. (2009). Acquisition of energy-efficient bipedal walking using CPG-based reinforcement learning. *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference*, pages 827–832.
- van Hasselt, H. and Wiering, M. a. (2007). Reinforcement Learning in Continuous Action Spaces. *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, (Adprl):272–279.
- Wiering, M. and van Otterlo, M. (2012). *Reinforcement Learning: State-Of-the-Art*. Adaptation, learning and optimization. Springer Berlin Heidelberg.
- Wikipedia (2013). Crawling (human): <http://en.wikipedia.org/wiki/Crawling>.
- Zhao, X., Zhang, J., and Qi, C. (2012). CPG and Reflexes Combined Adaptive Walking Control for AIBO. *2012 11th International Conference on Machine Learning and Applications*, pages 448–453.



# An Artificial Behavioral Immune System for cognitive robotics

Ignazio Infantino and Riccardo Rizzo

Institute of High Performance Computing and Networking (ICAR), National Research Council (CNR), Palermo, Italy  
 ignazio.infantino@cnr.it, riccardo.rizzo@cnr.it

## Abstract

The paper proposes the implementation of an artificial behavioral immune system (ABIS) that could be integrated on a robotic platform. The ultimate goal is to have a form of perceptual processing to create the sensation of disgust (on more in general of discomfort) that may influence the behavior that is determined by the cognitive modules. The present work, being oriented to human-robot interaction, is focused on perceptual processing of facial expression, and speech recognition. As regards the first aspect, the human face closer to the robot is detected, then facial features are extracted, and finally a vector of data which has an emotional valence linked to expression is determined. The speech is processed to recognize the individual words and the moods using a "sentiment analysis" technique. An artificial immune system is used to recognize positive and negative "feelings" that comes from interactions and generate the signal of discomfort. For training and validation we used a few human subjects and sequences of a fairy tale.

## Introduction

The complex human immune system does not act only when it is attacked by pathogens, but it enables behavioral mechanisms aimed at preventing potentially dangerous situations. Some psychological processes are activated by the so-called behavioral immune system (Schaller, 2011; Schaller and Park, 2011), with the aim to anticipate and avoid risky situations. This system is essentially based on processing of perceptual cues (Stewart, 1993) and provides answers that influence emotions, behavioral impulses, and other cognitive mechanisms. The behavioral immune system has a direct influence on the social interactions of humans, and in particular can have a significant impact on: social gregariousness (or the opposite extraversion); prejudices, discriminatory or unsociable behaviors; mate preferences and mating behaviors; normative and counter-normative behavior. Regardless of the mechanisms of high-level cognition, binding between the behavioral immune system and the feeling of disgust and discomfort seems to be quite shared and accepted. Such mechanism, it is suitable for a possible experimentation "in silico", in an artificial agent such as a robot to carry out behaviors more similar to humans (Hutchison, 2012).

Among other bio-inspired computational models, the endocrine system (Timmis et al., 2009; Guorui, 2013; Pérez et al., 2012) and swarm computing (Timmis et al., 2010) have some similarities and common aspects with the model of the immune system, although for the moment lacking extensive experiments that will enable to compare them.

The functioning of the human immune system is the subject of study and research, not only in the biomedical field, but also in the field of computer science since it shows capacity of adaptation, learning and memory, and represents a model that produces cognitive functions like the neurons (Stewart, 1993).

Artificial immune system models used in data processing are rough simplifications of the immune network theory proposed by Jerne (1974). All these model are based on the binding between antigen (input data) and antibody of different kinds (potential solutions). Other aspects of the mechanism of the immune system, captured by the majority of the models, are clonal expansion, affinity maturation, apoptosis and clonal suppression (Castiglione et al., 1999; Cutello and Nicosia, 2006; Pavone et al., 2012).

In general, computational models of the immune system can be used as learning classifier systems (Vargas et al., 2003b,a; Castiglione et al., 2001; Cutello and Nicosia, 2002). In robotics, artificial immuno-genetic networks have been tested with success in various contexts (Raza and Fernandez, 2012); behavior arbitration in mobile robots (Ishiguro et al., 1995; Mochida et al., 1995); error detection (Canham et al., 2003); robot cooperation (Dioubate et al., 2008); short-term learning (Whitbrook et al., 2008); trajectory generation (Acosta et al., 2010).

One of the most used Artificial Immune Systems (AIS) model in data processing is aiNet, proposed in de Castro and Zuben (2000). This model can be considered as a modified evolutionary algorithm (Stibor and Timmis, 2007) and was adapted to many applications, for example in Ciesielski et al. (2006) the aiNet was used for text document classification. A similar problem was also faced in Stibor and Timmis (2007), where the aiNet algorithm was used to "compress" a set of input pattern distributions, in order to understand if it

is possible to use the developed aiNet in place if the original input pattern distribution.

The aiNet was also using for biclustering in de Castro and de França (2007), and in optimization problems as reported in Tsang and Lau (2012).

The paper proposes the implementation of an artificial behavioral immune system (ABIS) that could be integrated on a robotic platform. The ultimate goal is to have a form of perceptual processing to create the sensation of discomfort that may influence the behavior that is determined by the cognitive modules. In a previous work we have addressed the problem of creating a sort of artificial proprioceptive system and suitable for a humanoid robot, and we have achieved a cognitive capacity for introspection (Infantino et al., 2013). In the present work, being oriented human-robot interaction (Prado et al., 2012), we focus our perceptual processing on two important aspects: the detection of facial expression, and the speech recognition. As regards the first aspect, the human face closer to the robot is detected, then facial features are extracted, and finally a vector of data which has emotional valence linked to expression is determined (Breazeal, 2003). We evaluate the dynamic evolution of this signal for few seconds. The speech instead is processed to recognize the individual words, then it follows a "sentiment analysis" (Pang and Lee, 2008; Cambria et al., 2013) attaching a mood. As previous, some numerical parameters are mediated over a given time interval.

The artificial immune system is used to distinguish between pleasant and not-pleasant situations. Such discrimination arises from the distinction between self and not-self cells that is typical of the biological immune system, and it comes out from a training phase during the first stage of the life.

### The Artificial Immune System

The aiNet algorithm is one of most used and effective models (de Castro and de França, 2007). The most important part of the aiNet algorithm is a list of simulated antibody cells (Ab-Cells) that interact with the simulated antigens (Ag-Cells) in an  $R^n$  space: both have a pattern vector  $\vec{w}_{Ab}$  and  $\vec{w}_{Ag}$ . The Euclidean distance between  $\vec{w}_{Ab}$  and  $\vec{w}_{Ag}$  is used to represent the binding or the affinity among cells.

### The Simulated Antibody Cell

The antibody cell properties can explain a lot about the working of the whole immune system. In a simulated artificial immune system the antibody cell could be an object constituted by a vector and some behaviours. The template that characterize the antibody is called paratope, and it can match a specific part of the antigen called epitope. In the aiNet model these parts of the antibody are simulated using real vectors  $\vec{w}_{Ab}$  and  $\vec{w}_{Ag}$  and this match is measured using the euclidean distance:

$$dist(\vec{w}_{Ab}, \vec{w}_{Ag}) \quad (1)$$

this distance is a factor that modulates the mutation of the cell and the amount of generated clones.

Like the biological model, the mutation of the Ab-Cell is aimed to increase the match with the Ag-Cell so that the amount of mutation is lower if the match is high. The mutation is obtained by randomly change the value of components of the real vector that represents the paratope.

A cell that has an high match with the Ag-Cell generates a lot of clones in order to increase the response of the immune system to the antigens so that the distance in eq. 1 modulates also the number of clones produced from an Ab-Cell.

### The aiNet algorithm

The aiNet algorithm, as discussed in Stibor and Timmis (2007), is reported in algorithm 1 By mean of this algorithm, we identify the antibody cells with its weight set  $w_{Ab}$ , called  $\vec{b}$  for sake of simplicity.

In the aiNet model when an Ag-Cell is presented to the AIS the nearest Ab-Cells activates and are subject to duplication and mutation (clonal expansion). The mutated cells more similar to the input Ag-Cell are maintained, while the other generated cells are killed (during a process that is called affinity maturation). The selected Ab-Cells have a chance to be part of the "memory" of the system: a set of conserved cells that are used in the case of a new infection. This long term memory is represented by the cell set B in algorithm 1, while the sets C and M are a working memory, i.e. the set of cells created during antigenic presentation.

### AIS and self-not-self recognition

One of the interesting characteristics of the immune systems is that the antibodies are capable to distinguish the antigen cells from the cells of the host organism. This feature is called *self-not-self recognition*. According to Farmer et al. (1986) antibody cells that attack the organism are produced during clonal expansion but they are killed, by other antibodies, during the clonal suppression phase.

This mechanism requires a training phase aimed to recognize possible dangerous antibodies and how to eliminate them. Like antigens all the antibodies have an epitope that can be recognized by other antibodies, these epitopes are used during clonal suppression. Dangerous antibodies are recognized by using their epitopes and removed by other antibodies (Farmer et al., 1986). The training phase that allows to recognize the dangerous antibodies take place during the development of the embryo of the organism.

This model can be implemented in the artificial immune system by preceding the training session with a initial learning that involved only self cells: these cells are presented to the artificial immune system so that it can develop the antibodies that recognize these self cells. In fig. 1 and fig. 2 a toy example better explains self-not-self recognition: the input data are in fig. 1 and the self data, the blob on the top-left, are submitted to the system during the first part of the

**Algorithm 1** The aiNet algorithm

- 1: Random initialize a set of Ab-Cells with the vector templates  $B = \{b_1, b_2, \dots, b_N\}$
- 2: **while**  $t < t_{max}$  **do**
- 3:   **for** each antigen  $ag \in G$  with the vector template  $\vec{g}$  **do**
- 4:     split  $B$  in two parts,  $C$  and  $T$ ,  $|C| = n$ ,  $(C \cup T = B, C \cap T = \emptyset)$  by putting in  $C$  the  $n$  antibodies with minimum distance from  $\vec{g}$

$$C = \{\vec{c}_i \mid \text{dist}(\vec{c}_i, \vec{g}) < \text{dist}(\vec{t}_i, \vec{g}) \mid \forall \vec{c}_i \in C, \forall \vec{t}_j \in T\} \quad (2)$$

- 5:     **clonal expansion:** clone the cells in  $C$  according with their distance  $\text{dist}(\vec{c}_i, \vec{g})$
- 6:     mutate the cells  $c_i \in C$

$$\vec{c}_i = \vec{c}_i + \alpha * (\vec{g} - \vec{c}_i) \quad (3)$$

where  $\alpha = \frac{r}{\text{dist}(\vec{g}, \vec{c}_i)}$ ,  $r \in [0, 1]$

- 7:     split  $C$  in two parts,  $M$  and  $T_2$ ,  $(M \cup T_2 = C, M \cap T_2 = \emptyset)$  by putting in  $M$  the antibodies with

$$M = \{\vec{m}_i \mid \text{dist}(\vec{m}_i, \vec{g}) < \text{dist}(\vec{t}_i, \vec{g}) \mid \forall \vec{m}_i \in M, \forall \vec{t}_j \in T_2\} \quad (4)$$

with  $|M| = \psi * |C|$ ,  $|T_2| = (1 - \psi) * |C|$ ,

- 8:     **apoptosis:** remove the memory clones in  $M$  with  $\text{dist}(\vec{m}_i, \vec{g}) > \sigma_d$
- 9:     **clonal suppression:** remove clones  $m_j$  in  $M$  with  $\text{dist}(\vec{m}_j, \vec{m}_i) < \sigma_s$  with  $i \neq j$
- 10:     $B \leftarrow B \cup M$
- 11: **end for**
- 12:    remove from  $B$  all the cells  $b_j$  with  $\text{dist}(\vec{b}_j, \vec{b}_i) < \sigma_s$  with  $i \neq j$
- 13:    create new random antibodies in  $B$
- 14:     $t \leftarrow t + 1$
- 15: **end while**

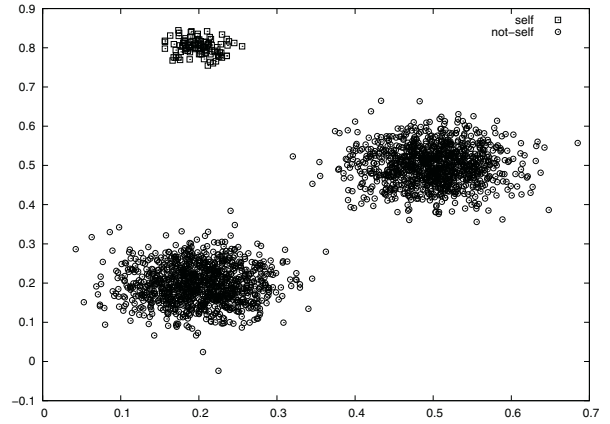


Figure 1: A toy self-not-self problem. The pattern distribution top-left represent the self pattern; these patterns are presented to the AIS during the first step of the learning phase.

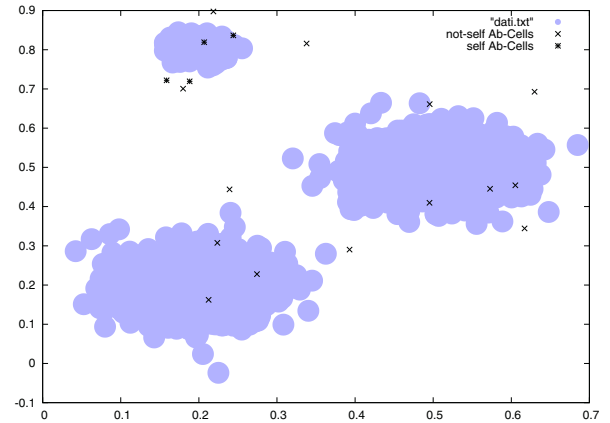


Figure 2: After the learning phase the AIS is ready to recognize self and non-self data. The antibodies not-self near the self Ab-Cells will be destroyed during the clonal suppression.

training session.

The AIS reacts creating a set of antibodies, in the following called *self antibodies*, that recognize the self cells. These antibodies are placed in the set  $B$ , the long term memory of the system. During the normal operation of the AIS antibody cells are produced: if these cells are near self antibodies they should be destroyed and a signal is generated. This mechanism is shown in fig. 3.

Using this schema it is possible to recognize a threat to the system because it is the one that generate antibodies of the type non-self, while any other organism cell will generate antibodies of the self kind, that will be immediately destroyed.

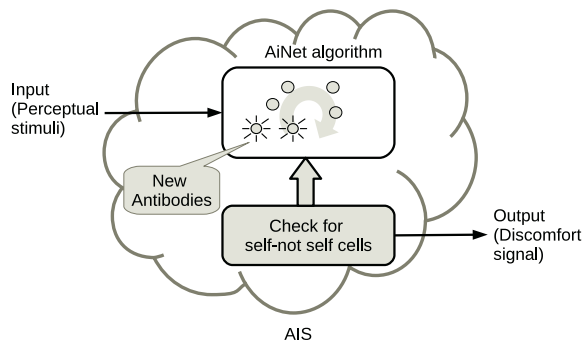


Figure 3: The structure of the system for self-not-self recognition. A signal is generated when antibodies self are activated.

### The Artificial Behavioral Immune System

The implemented system has been tested on the robotic platform NAO, and is an integral part of the behavioral system in the context of human-humanoid robot. The behavior comes from a cognitive architecture based on the model PSI (Cai et al., 2011), which in the past we have enriched with introspective capabilities (Infantino et al., 2013), with links between visual perceptions and emotions (Gaglio et al., 2011; Infantino et al., 2012; Infantino, 2012), and developing some creative capabilities (Augello et al., 2013b) by binding different perceptual representations. We have chosen two important sensorial stimuli that involve both high-level cognitive functions, and behavioral reactions influenced by the artificial immune system. Speech and facial expressions are two basic channels of communication in social interactions (see for example Prado et al. (2012), and Breazeal (2003)) both on a subconscious and rational level. The proposed system focuses on the detection and recognition of specific patterns from such perceptual stimuli that have a significant emotional impact (see fig. 4) The detection of words, the prosody, the tone of voice, and so on are all factors that influence social interaction (Pennebaker et al., 2003). The target robotic platform used for our experiments has a speech recognizer with detection of individual words and phrases. We have drawn upon an approach that mixes sentiment analysis and opinion mining (Cambria et al., 2013): to define the emotional content of a text, it is established a correspondence between word and five parameters: pleasantness, attention, sensitivity, aptitude, and polarity. The values taken by these parameters constitute the first part of the input of the immune system, and we consider it stored in long term memory by affective system. (see fig. 5). In a similar manner, we use some parameters extracted from facial expressions that typically allow you to classify it, and that are linked to patterns stored in the visual long term memory. Putting to-

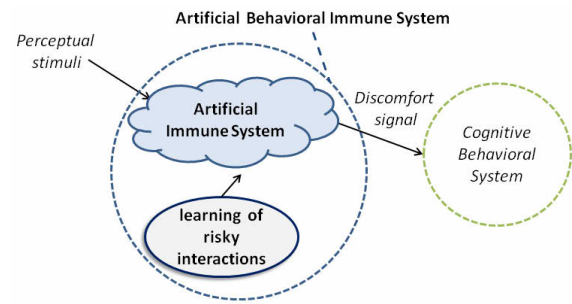


Figure 4: ABIS-schema

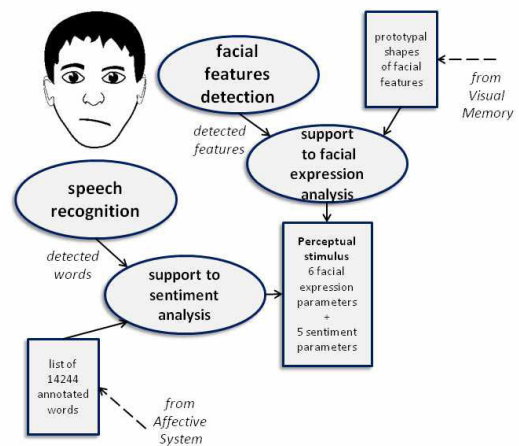


Figure 5: Perceptual processing

gether these two (processed) sensorial inputs, the immune system is trained to recognize risk situations. When a pattern is recognized potentially dangerous generates an alert that signals to activate typical behaviors of a uncomfortable situation. Of course, the cognitive behavioral system will be affected by this alert, but this could be ignored or hidden by the high-level planning and reasoning (see fig. 4).

### Speech sentiment analysis

The experimental setup is constituted by a human subject that directly speaks to the robot. The robot reacts emotionally based on the content of the speech and facial expressions detected in sync. As previously mentioned, the emotional content of speech is characterized by five numerical parameters resulting from publicly available semantic and affective database (MIT-Media-Laboratory, 2013). The database used contains 14244 concepts, and it is possible queried it by web. For example, the page <http://sentic.net/api/en/concept/smile/> corresponds to a xml file that contains the related concepts (sneer, start-laugh, hear-joke, smile-laugh, giggle), and numerical values of the parameters (pleasantness = 0.997, attention = 0.0, sensitiv-



ity = 0.0, aptitude = 0.0, polarity = 0.332).

## Face expression analysis

To characterize facial expression we used the six coefficients of Animation Units (AU) by using an API widely used in the context of face tracking (Microsoft, 2013). They are a subset coefficients defined in the Candide3 model (Ahlberg, 2001) that uses 87 2D points on the face to track a human head. From these six values it is possible to classify the facial expression considering seven basic expressions: neutral face, upper lip raised, jaw lowered, lip stretched, brow lowered, lip corner depressed and outer brow raised. The range of these coefficients is between -1.0 and 1.0. For example, the first coefficient indicate the lips movement, the value +1 indicates the lips completely opened and -1 completely closed. The afore-mentioned six coefficients can be used to classify an emotional state through a series of IF-THEN rules and a set of threshold values.

## Experiments

To validate the proposed approach we have designed an experiment that seems simple: the robot listens to a fairy tale and observes the facial expressions of the narrator. The data collected in this phase are used to train the immune system by means of a semi-manual labeling. A fairy tale being aimed at children, typically uses simple words and facial expression in the narrative is essential to strengthen the emotional content of the story (Alm et al., 2005). As with children, the robot can learn from a fairy tale a behavioral and emotional model on which to base its cognitive development. The learned model, in our case via the artificial immune behavioral system, can be employed to deal with normal interactions with humans. In the testing phase in fact a human speaks of a farm, a vegetable garden, animals, arousing emotions to the robot derived from having listened to the famous *The Tale of Peter Rabbit* by Beatrix Potter. The robot watched a video of this tale available on the web (Socratica, 2013). The narrative is divided in 35 sequences, each corresponding to a part of speech and a sequence of facial features detected (see fig. 6).

Figure 7 shows the maximum, minimum, and average (blue dashed line) of the parameters of sentiment for the 35 sequences.

For the learning phase of the six AU parameters derived from facial features are grouped by similarity (using a simple k-means), and clusters are labeled manually as negative or positive inputs. Figure 8 shows the average values of the Animation Units derived from facial features detected for the 35 sequences.

Preliminary experiments show a rate of correct recognition of discomfort situations of about 70%.

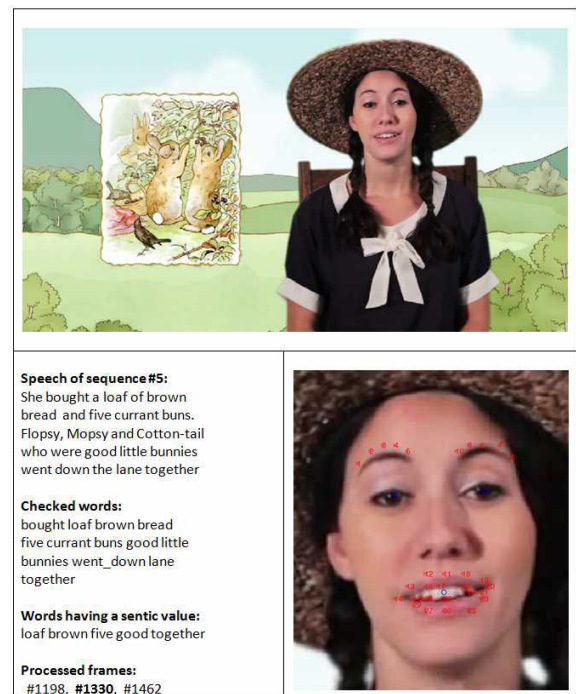


Figure 6: Perception example of sequence 5 of 35

## Conclusions and future work

The robot as a child could learn how to recognize positive and negative situations by visual and voice interaction with a tutor. In the article we proposed a low level mechanism that activates the alarm signals relating to the discomfort. It is inspired by the human immune system, and is based on the influences that such a system has on behavior.

We are conducting more extensive experimentation that allow us to understand how to improve the artificial immune system, integrating other perceptual signals, and optimizing the learning phase.

## Acknowledgements

We thanks Federico Corrao for the experimental setup. Financial support of the research is partially given by Ministero dello Sviluppo Economico e Innovazione (MISE), bando "MADE IN ITALY", project MI01 00424.

## References

- Acosta, G. G., León, J. F., and Mayosky, M. A. (2010). Artificial immune system inspired behavior coordination for autonomous mobile robot trajectory generation. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–6. IEEE.
- Ahlberg, J. (2001). TCANDIDE-3 – an updated parameterized face. Technical report, Dept. of Electrical Engineering, Linköping University, Sweden. <http://www.icg.isy.liu.se/candide>.

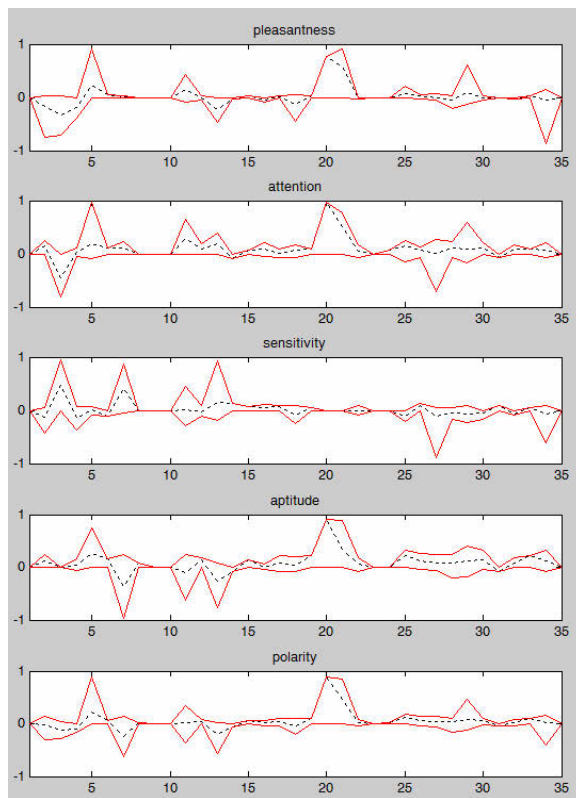


Figure 7: Sentic values of the 35 sequences of video used in the experiments

Alm, C. O., Roth, D., and Sproat, R. (2005). Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 579–586. Association for Computational Linguistics.

Augello, A., Infantino, I., Pilato, G., Rizzo, R., and Vella, F. (2013a). Binding representational spaces of colors and emotions for creativity. *Biologically Inspired Cognitive Architectures, Special Issue of the BICA 2012 Meeting*, in printing.

Augello, A., Infantino, I., Pilato, G., Rizzo, R., and Vella, F. (2013b). Introducing a creative process on a cognitive architecture. *Biologically Inspired Cognitive Architectures*, available on line at: [www.sciencedirect.com/science/article/pii/S2212683X13000467](http://www.sciencedirect.com/science/article/pii/S2212683X13000467).

Breazeal, C. (2003). Emotive qualities in lip-synchronized robot speech. *Advanced robotics*, 17(2):97–113.

Cai, Z., Goertzel, B., and Geisweiller, N. (2011). Openspi: Realizing dorners psi cognitive model in the opencog integrative agi architecture. In *Artificial General Intelligence*, pages 212–221. Springer.

Cambria, E., Mazzocco, T., and Hussain, A. (2013). Application of multi-dimensional scaling and artificial neural networks for biologically inspired opinion mining. *Biologically Inspired Cognitive Architectures*, 4(0):41 – 53.

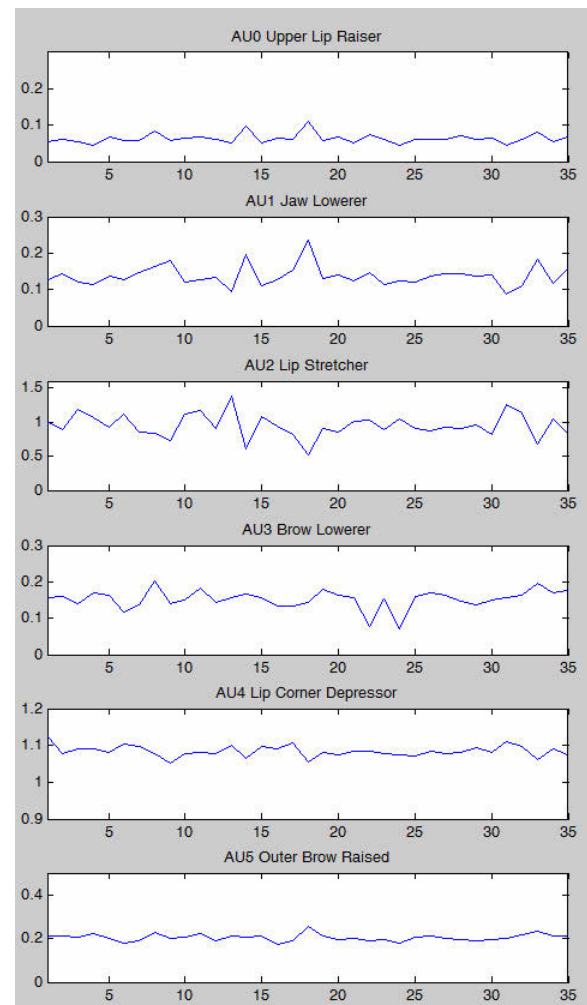


Figure 8: Animation unit values of the 35 sequences of video used in the experiments

Canham, R., Jackson, A. H., and Tyrrell, A. (2003). Robot error detection using an artificial immune system. In *Evolvable Hardware, 2003. Proceedings. NASA/DoD Conference on*, pages 199–207. IEEE.

Castiglione, F., Mannella, G., Motta, S., and Nicosia, G. (1999). A network of cellular automata for the simulation of the immune system. *International Journal of Modern Physics C*, 10(04):677–686.

Castiglione, F., Motta, S., and Nicosia, G. (2001). Pattern recognition by primary and secondary response of an artificial immune system. *Theory in Biosciences*, 120(2):93–106.

Ciesielski, K., WierzchoÅ, S., and KÅopotek, M. (2006). An immune network for contextual text data clustering. *Artificial Immune Systems*, pages 432–445.

Cutello, V. and Nicosia, G. (2002). Multiple learning using immune algorithms. In *Proceedings of 4th International Conference on Recent Advances in Soft Computing, RASC*, pages 102–107.

- Cutello, V. and Nicosia, G. (2006). A clonal selection algorithm for coloring, hitting set and satisfiability problems. In *Neural Nets*, pages 324–337. Springer.
- de Castro, L. N. and Zuben, F. V. (2000). An Evolutionary Immune Network for Data Clustering. In *IEEE SBRN Brazilian Symposium on Artificial Neural Networks*, pages 84–89.
- de Castro, P. and de França, F. (2007). Applying biclustering to text mining: an immune-inspired approach. In de Castro, L. N., Zuben, F. J. V., and Knidel, H., editors, *Artificial Immune Systems*, pages 83–94. Springer Berlin Heidelberg.
- Dioubate, M., Guanzheng, T., and Toure Mohamed, L. (2008). An artificial immune system based multi-agent model and its application to robot cooperation problem. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 3033–3039. IEEE.
- Farmer, J., Packard, N., and Perelson, A. (1986). The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena*, pages 187–204.
- Gaglio, S., Infantino, I., Pilato, G., Rizzo, R., and Vella, F. (2011). Vision and emotional flow in a cognitive architecture for human-machine interaction. *Frontiers in Artificial Intelligence and Applications*, 233:112–117.
- Guorui, H. (2013). The emotional robot model based on endocrine system. In *Intelligence Computation and Evolutionary Computation*, pages 633–645. Springer.
- Hutchison, W. R. (2012). The central role for behavior analysis in modern robotics, and vice versa. *The Behavior Analyst*, 35(1):29.
- Infantino, I. (2012). Affective human-humanoid interaction through cognitive architecture. In *The Future of Humanoid Robots - Research and Applications, Dr. Riadh Zaier (Ed.)*, pages 147–164. InTech.
- Infantino, I., Pilato, G., Rizzo, R., and Vella, F. (2012). I Feel Blue: Robots and Humans Sharing Color Representation for Emotional Cognitive Interaction. *Biologically Inspired Cognitive Architectures 2012, Advances in Intelligent Systems and Computing*, 196:161–166.
- Infantino, I., Pilato, G., Rizzo, R., and Vella, F. (2013). A practical approach to humanoid introspection. *International Journal of Advanced Robotic Systems. Topic: Human Machine Interaction*, 10:246–256.
- Ishiguro, A., Kondo, T., Watanabe, Y., and Uchikawa, Y. (1995). Dynamic behavior arbitration of autonomous mobile robots using immune networks. In *Evolutionary Computation, 1995., IEEE International Conference on*, volume 2, pages 722–727. IEEE.
- Jerne, N. (1974). Towards a network theory of the immune system. *Annales d'immunologie*, 125C:373–389.
- Microsoft (2013). Face tracking. <http://msdn.microsoft.com/en-us/library/jj130970.aspx> (last access on May 2013).
- MIT-Media-Laboratory (2013). SenticNet. <http://www.sentic.net> (last access on May 2013).
- Mochida, T., Ishiguro, A., Aoki, T., and Uchikawa, Y. (1995). Behavior arbitration for autonomous mobile robots using emotion mechanisms. In *Intelligent Robots and Systems 95: Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 3, pages 516–521. IEEE.
- Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.
- Pavone, M., Narzisi, G., and Nicosia, G. (2012). Clonal selection: an immunological algorithm for global optimization over continuous spaces. *Journal of Global Optimization*, 53(4):769–808.
- Pennebaker, J. W., Mehl, M. R., and Niederhoffer, K. G. (2003). Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54(1):547–577.
- Pérez, C. H., Escribano, G. S., and Sanz, R. (2012). The morpho-functional approach to emotion modelling in robotics. *Adaptive Behavior*, 20(5):388–404.
- Prado, J. A., Simplício, C., Lori, N. F., and Dias, J. (2012). Visuo-auditory multimodal emotional structure to improve human-robot-interaction. *International Journal of Social Robotics*, 4(1):29–51.
- Raza, A. and Fernandez, B. R. (2012). Immuno-inspired robotic applications: a review. *arXiv preprint arXiv:1202.4261*.
- Schaller, M. (2011). The behavioural immune system and the psychology of human sociality. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1583):3418–3426.
- Schaller, M. and Park, J. H. (2011). The behavioral immune system (and why it matters). *Current Directions in Psychological Science*, 20(2):99–103.
- Socratica (2013). The Tale of Peter Rabbit by Beatrix Potter. <http://www.youtube.com/user/SocraticaKids?v=RE-znzH6oro> (last access on May 2013).
- Stewart, J. (1993). Cognition without neurones: adaptation, learning and memory in the immune system. *Cognitiva*, 5(2):187–202.
- Stibor, T. and Timmis, J. (2007). An Investigation on the Compression Quality of aiNet. *2007 IEEE Symposium on Foundations of Computational Intelligence*, pages 495–502.
- Timmis, J., Andrews, P., and Hart, E. (2010). On artificial immune systems and swarm intelligence. *Swarm Intelligence*, 4(4):247–273.
- Timmis, J., Neal, M., and Thorniley, J. (2009). An adaptive neuro-endocrine system for robotic systems. In *Robotic Intelligence in Informationally Structured Space, 2009. RIISS'09. IEEE Workshop on*, pages 129–136. IEEE.
- Tsang, W. and Lau, H. (2012). Clustering-Based Multi-Objective Immune Optimization Evolutionary Algorithm. In *Artificial Immune Systems*, pages 72–85.

- Vargas, P. A., de Castro, L. N., Michelan, R., and Von Zuben, F. J. (2003a). Implementation of an immuno-genetic network on a real khepera ii robot. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 1, pages 420–426. IEEE.
- Vargas, P. A., de Castro, L. N., and Von Zuben, F. J. (2003b). Mapping artificial immune systems into learning classifier systems. In *Learning Classifier Systems*, pages 163–186. Springer.
- Whitbrook, A., Aickelin, U., and Garibaldi, J. (2008). An idiotypic immune network as a short-term learning architecture for mobile robots. In *Artificial Immune Systems*, pages 266–278. Springer.



# Evolutionary Explanations for Spatial Language - A Case Study on Landmarks

Michael Spranger

Sony CSL Paris, 6, rue Amyot, 75005 Paris, France  
 Sony CSL Tokyo, 3-14-13 Higashi Gotanda, 141-0022 Tokyo, Japan  
 spranger@csl.sony.fr

## Abstract

This paper reports on latest results from a larger research program trying to understand the computational principles behind the processing, acquisition and evolution of spatial language. Here we explore an evolutionary explanation of spatial landmarks by carrying out detailed computational experiments. The paper discusses the necessary mechanisms for representing different strategies involving landmarks, their effect on success in communication and their impact on the formation of spatial relations.

## Introduction

A lot of agent-based models examining evolution of language are dealing with (single-word) lexical systems such as the formation of colour terms (Steels and Belpaeme, 2005). There are some examples that try to go further and focus on the evolution of grammar, such as van Trijp (2008). However, those approaches typically start from predefined lexicons. The same holds for spatial language where early work has focussed on lexicons (Steels, 1995) and recent work has touched upon grammar (Spranger and Steels, 2012). This paper presents intermediate work that tries to bridge the gap between purely single-word lexical systems and more grammatical systems where words take on particular functions.

We are examining locative phrases consisting of spatial relations and landmarks. Here is an example of such a phrase from German.

(1) *der*        *Block*        *rechts*        *der*  
       the.NOM block.NOM right.PREP the.GEN  
*Kiste*  
       box.GEN  
       ‘The block to the right of the box’.

The phrase encodes a specific way of construing reality by integrating a number of spatial *strategies*. The phrase combines the *projective* spatial relation “rechts” (right) with a landmark – an inanimate object denoted by the phrase “der Kiste” (the box). The speaker explicitly *marked* the landmark (not obligatory).

The precise usage of landmarks in a language is culturally negotiated. In English inanimate objects can be used as landmarks. Other languages restrict landmarks to animate objects or only humans, and in some cases just to trees (Levinson (2003)). Moreover, the strategies for expressing conceptual choices – lexicon and grammar – differ across languages (Svorou (1994)).

Cultural diversity of spatial language strategies is the prime argument for cultural evolution of language (Evans and Levinson, 2009). Recently, a number of attempts were made at explaining the cultural evolution of spatial language such as toponyms (Schulz et al., 2006), the role of perspective – “me” and “you” – (Steels and Loetzsch, 2008), the origins of conceptualization strategies (Spranger, 2011) and grammar (Spranger and Steels, 2012). In this paper we focus on the role of landmarks including perspective but also inanimate objects. As the methodological framework we follow standard methodology for evolutionary explanations in biology (Tinbergen, 1963). In particular, we try to answer questions about processing, function, acquisition and evolution of spatial landmark systems.

## Experimental Setup

We employ a *language game* paradigm (Steels, 2012) in order to study spatial language use and evolution. Two robots from a population are trying to draw each others attention to objects in the environment using language. Figure 1 shows an example setup.

The environment consists of a number of blocks of equal size and color (circles), boxes (rectangle) and interlocutors (arrows). The vision system of each robot tracks objects in the vicinity and establishes a model of the environment consisting of blocks (circles) with real-valued distances and orientations of objects with respect to the body of the robot. The environment is open-ended. New blocks, boxes and robots are added or removed and their spatial configuration is changed.

1. First the robots establish a joint attentional frame (Tomasello (1995)). Subsequently, each agent scans the environment and establishes a situation model of the present

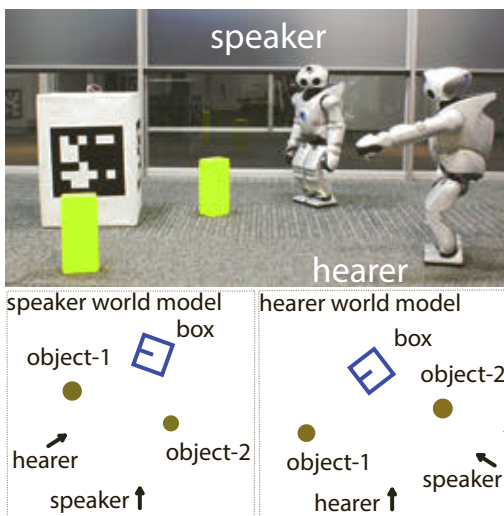


Figure 1: This figure shows the set-up for spatial language game experiments. The images left and right show the internal situation model as perceived by each robot.

objects and their position.

2. One of the robots is randomly assigned the role of speaker. The speaker randomly selects an object from the situation model (further called the topic  $T$ ). The speaker tries to find a discriminating spatial relation applied to a particular landmark or perspective for describing  $T$ . Subsequently, the speaker looks up the words he associates with the spatial relation and (possibly) the landmark and produces an utterance. For instance, he might say “links kiste” (roughly: left of the box) if the robots operate a German lexicon<sup>1</sup>.
3. The hearer looks up which relation and landmarks is associated with the words in the utterance. He examines his situation model to find out whether there is a unique object which satisfies the spatial relation with respect to the landmark. The hearer then points to this object. If no landmark is specified, the hearer might try different possible landmarks and see if there is a combination of spatial relation with landmark that is discriminating an object best.
4. The speaker checks whether the hearer points to  $T$ . If the hearer pointed correctly the game is a success and the speaker signals this outcome to the hearer. If the game is a failure, the speaker points to the topic  $T$ .

<sup>1</sup>Notice that there is no grammar in this phrase. Also, the landmark can be omitted.

Obviously things can go wrong in these interactions especially if the agents are in the process of building a language to succeed in these interactions. The speaker might not have an appropriate spatial relation to discriminate the topic from the rest of the objects in the situation model. Possibly, he has no word for denoting the landmark he has in mind which can lead to confusion with other landmarks and consequently with other objects in the situation model. On the hearer side, the robot maybe does not know certain words in the utterance. Lastly, the hearer might point to the wrong object. These situations present opportunities to update in particular to acquire new words and also expand internal language representations.

## The Role of Landmarks in Spatial Language

To give an evolutionary explanation of a phenomenon entails to give an account of the role the trait plays for the evolutionary success. For us, *communicative success* is the primary selective force in cultural language evolution. Consequently, we have to identify the role of landmarks with respect to that evolutionary pressure.

We use spatial language games to measure what is the impact of landmark systems on communicative success. Populations of agents are given different language and conceptualization strategies and we measure their performance in multi-agent simulations. In all populations agents are using *projective* spatial relations equivalent to English “front”, “back”, “left” and “right”. The relations are defined as a similarity functions which are computed from the difference in angles between an object and the prototypical angle of the category (Spranger, 2012). We compare 5 populations.

**egocentric (ego)** Agents use a single spatial relation in each utterance which is always interpreted egocentric to each robot. For instance, if the speaker conceptualizes “left” to refer to an object, the hearer will interpret the relation from his perspective.

**perspective, unmarked (pp, um)** Agents use the perspective of themselves but also of the other robot (Steels and Loetzsch, 2008, similar to). Upon hearing the term “left” the hearer retrieves the best possible interpretation taking into consideration his own and the other robot’s perspective on the environment. Similarly, the speaker might use an utterance such as “left” but refer to a position left of the hearer rather than left of himself.

**perspective, marked (pp, m)** Agents are using different perspectives but also clearly marking which perspective was used. For example, the speaker might say “left you” to indicate spatial relation and perspective.

**landmarks, unmarked (lm, um)** This population can use perspective (robots) and allocentric landmarks (the box), in conceptualizing reality. They do not communicate which object was used as a landmark.

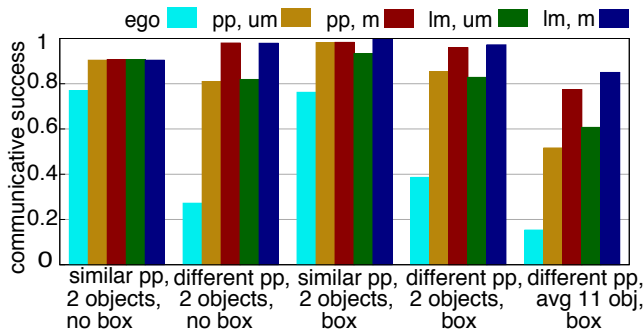


Figure 2: Results for populations of 10 agents tested in different experimental conditions. For each result agents interacted 2000 times. Every successful interaction counts with 1.0, otherwise with 0.0. These scores are averaged over all interactions.

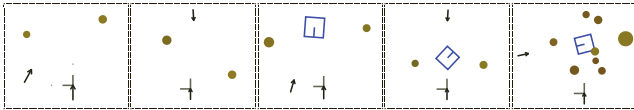


Figure 3: Examples of spatial scenes from each experimental condition ordered from left to right as in Figure 2. Left, for instance, agents (black arrows) share a similar perspective on the spatial scenes, there are always 2 objects per scene (circles) and no box. To the right the most difficult condition is shown. Agents have differing perspectives, there are on average 11 objects and there is a box.

**landmarks, marked (lm, m)** Same population as *landmarks, unmarked*. But, agents express which landmark they use in all utterances.

Figure 2 shows communicative performance of each population in different experimental conditions. Each condition consists of many spatial situations each of which include two robots and blocks whose positions change. But, the conditions differ in 1) whether a box is present, 2) the average number of objects and 3) the typical perspective of robots (see Figure 3).

The results show that agents capable of using different perspectives and landmarks perform better in communication than agents without. The effect increases the more objects are in each spatial context and the more the viewpoints of the interlocutors on the scene diverge. This is because agents which can use different landmarks are much more flexible. They have more choice of relation-landmark pairs and often can choose one that fits best or better for discrimination. There are three important observations. First, if the environment is simple and agents have a similar perspec-

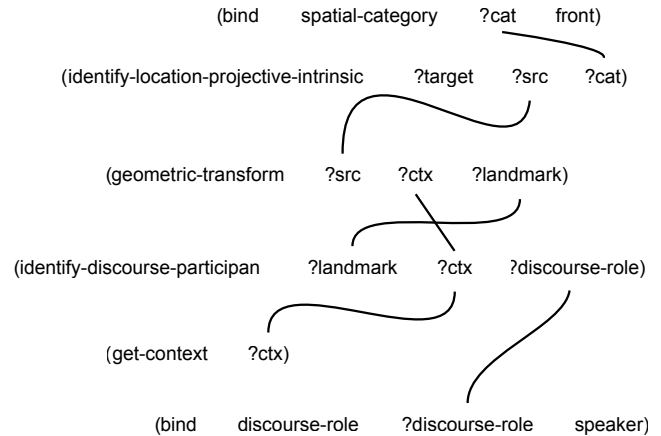


Figure 4: Representation for a speaker centric conceptualization.

tive on the scene, it is sufficient to use egocentric interpretation of phrases. Second, the best strategy is to use any landmark possible (both robots and allocentric landmarks). Third, marking which landmark or perspective was used always outperforms the unmarked case in which agents use different landmarks but cannot express them.

## Representing Spatial Strategies

We use a computational formalism called *Incremental Recruitment Language* (IRL) to represent the different strategies of conceptualising reality. IRL represents semantics as a constraint program. Programs consist of pointers to data items and operations on these data items. Figure 4 shows an example of such programs for a speaker centric conceptualisation such as in “vor mir” (in front of me). The program consists of pointers to the category *front*, and the discourse role *speaker*. Both are introduced via so called *bind*-statements. The other items in the program are functions linked via variables (starting with ?). For instance, *get-context* introduces the situation model computed by the vision system.

Spatial relations are implemented following ideas in cognitive semantics (Herskovits, 1986) and prototype theory (Rosch, 1975). We implemented two types of categories, distance-based (proximal) and angle-based (projective and absolute). We only focus on projective categories in this paper. Projective categories have a focal region around a specified axis. Similarity of some location to an angular category depends on the distance of angles. For instance, the *front* category has a high degree of applicability along the frontal axis. The following equations defines the degree of applicability, i.e. similarity,  $\text{sim}_a \in [0, 1]$  given an object  $o$  and an angular category  $c$  and a parameter  $\sigma$  which steers

the steepness of the function.

$$\begin{aligned}\text{sim}_a(o, c) &:= e^{-\frac{1}{2\sigma_c} d_a(o, c)} \\ d_a(o, c) &:= |a_o - a_c|\end{aligned}$$

$a_o$  denotes the angle of the position of  $o$  to the coordinate center and  $a_c$  is the prototypical angle of  $c$ .

Besides spatial relations, the program also contains operations. Operations take input parameters and compute new output parameters.

`get-context` fetches the situation model from the vision system of the robot.

`identify-location-projective-intrinsic` takes a spatial relation and an input set and finds the single object that best fits the category.

`geometric-transform` takes an input set and a landmark and transforms the input set so that the spatial positions and orientations are seen from the landmark.

`identify-discourse-participant` takes a discourse role such as `speaker` and `hearer` and identifies the interlocutor denoted by the discourse role.

Earlier work has examined how aspects of these strategies can evolve autonomously in populations of agents (Spranger, 2011). In the experiments presented in this paper, agents are assumed to have such strategies. They are given to the agents.

## The Formation of Spatial Relation Systems

We can now examine which operators are necessary for the formation of spatial category systems. Given a strategy and a set of invention, adoption and alignment operators concrete systems of spatial relations arise in populations of agents (Spranger, 2012). The following paragraphs detail the operators.

**Invention:** *Speaker cannot find a discriminating spatial category in production*

- **Diagnostic:** When the speaker cannot conceptualize a meaning (step 2 of the spatial language game fails).
- **Repair:** The speaker constructs a spatial relation  $R$  based on the relevant strategy (projective) and the topic pointed at. The new category is necessarily based on the distance or angle observed for the topic object (the initial sigma is small 0.1). Additionally, the speaker invents a new construction associating  $R$  with  $s$ .

**Adoption:** *Hearer encounters unknown spatial term  $s$*

- **Diagnostic:** When the hearer does not know a term (step 3 fails).
- **Repair:** The hearer signals failure and the speaker points to the topic  $T$ . The hearer then constructs a spatial relation  $R$  based on the relevant strategy and the topic pointed at. Additionally, the speaker invents a new construction associating  $R$  with  $s$ .

**Category alignment** Projective categories are represented by prototypical angles. After each interaction agents update the prototypical angle to better reflect the new observation by averaging the angles of objects in the sample set  $S$ . The new prototypical angle  $a_c$  of the category is computed using the following formula for averaging angles.

$$a_c = \text{atan2} \left( \frac{1}{|S|} \sum_{s \in S} \sin a_s, \frac{1}{|S|} \sum_{s \in S} \cos a_s \right) \quad (2)$$

The new  $\sigma$  value  $\sigma'$  which describes the shape of the similarity function of the category is adapted using the following formula.

$$\sigma'_c = \sigma_c + \alpha_\sigma \cdot \left( \sigma_c - \sqrt{\frac{1}{|S| - 1} \sum_{s \in S} (a_c - a_s)^2} \right) \quad (3)$$

This formula describes how much the new  $\sigma_c$  of the category  $c$  is pushed in the direction of the angle standard deviation of the sample set by a factor<sup>2</sup> of  $\alpha_\sigma \in [0, \infty]$ .

**Lexicon alignment** Its possible that a single agent can name the same category with different spatial terms. This phenomenon is called synonymy. Allowing agents to track synonymy in their lexicons can be beneficial for overall lexicon size, but only if agents have additional mechanisms for coping with it. Such a mechanism, called *lateral inhibition*, was introduced in Steels (1995):

- In case the interaction was a success both speaker and hearer reward the winning construction – the one used in production and interpretation – by a score of  $\delta_{\text{success}}$ . Competing constructions are punished by  $\delta_{\text{inhibit}}$ . There are two types of competing constructions. First, there are those constructions which associate the same spatial relation but with a different word. Second, there are constructions that link the same word to different spatial relations.
- After a failed game, both speaker and hearer decrease the score of the used association with  $\delta_{\text{fail}}$ .

<sup>2</sup> $\alpha$  is given by the experimenter and in all experiments described here  $\alpha = 0.5$



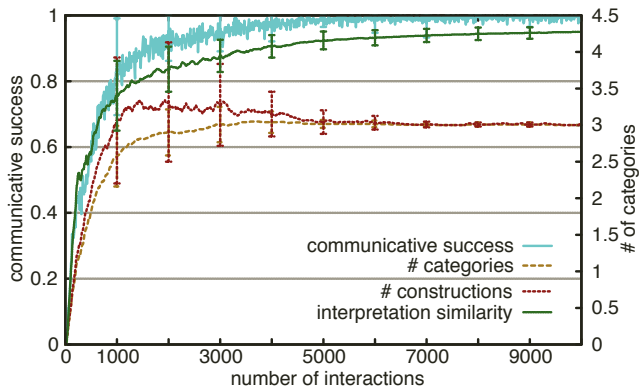


Figure 5: Results for a formation experiment in which agents develop a projective category system using a single landmark strategy (the box is used as landmark) in simple environments. Only few objects are present.

Figure 5 shows results for the formation of a projective categories. The population consists of 10 agents. an experimental run lasts 10000 interaction. Experimental runs are repeated multiple times and a number of measures tracked for confirming the convergence of the system.

**Communicative Success** tracks how successful the population is. After each interaction, it is checked whether speaker and hearer interacted successfully. Success is counted as 1.0, failure as 0.0.

**# Categories, # Constructions** is a measure for the average spatial relation inventory size and the number of constructions (number of words) known to the agent. These can diverge in case of synonymy and when there are words not denoting spatial relations but landmarks.

**Interpretation similarity** measures how similar the meaning of words are. Each agent tracks spatial relations without ever having access to the exact representation of all other agents. This measure compares the spatial relations of each agent based on how similar they are. 0.0 means no similarity. 1.0 would mean that all agents have exactly the same categories.

The results in Figure 5 show that agents can indeed go from no systems of spatial relations to one that is successful. In this case the population agrees on 3 categories on average. The little overshoot of constructions vs categories in the beginning is caused by synonymy that is later removed by the alignment operators.

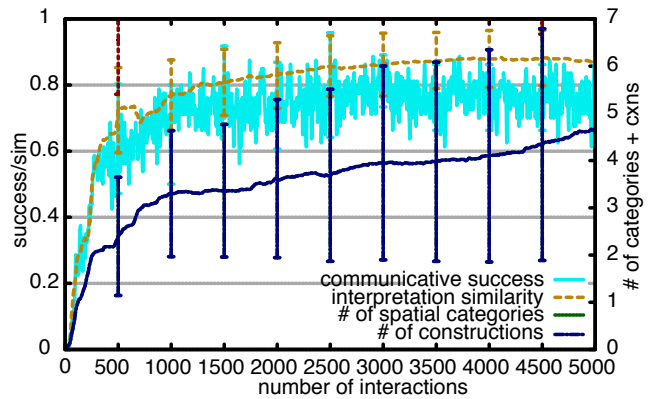


Figure 6: Dynamics of a population of 10 agents which develop a spatial category system (egocentric strategy) of 5 spatial relations in the *similar pp, 2 obj, no box* condition (5000 interactions, 8 trials). At the same time the spatial relations align across the population (see rise in *interpretation similarity*).

## The Formation of Spatial Relation Systems and Landmarks

Next, we apply the insights from category formation and systematically study the effect of landmarks on such systems. We re-use the populations introduced in the previous section. All spatial relations (and terms) are removed from agents and instead they are provided with the language change operators described in the previous section.

Figure 6 shows the dynamics of these operators over time. Agents start without spatial relations and spatial terms. Gradually they invent and align their linguistic repositories. At the same time the population becomes more successful in communication. Figure 7 compares different populations in a number of experimental conditions (same as used earlier). When agents do not have the means to use landmarks they fail in developing successful systems of spatial relations (if the environment is complex). Again the populations which can express the landmark they are using are outperforming agents which cannot. Agents which can use the full scope of landmarks (me/you/box) available in scenes outperform those which can only use perspective (me/you) which outperform those that can only use an egocentric strategy.

Notice, how *landmark, unmarked* performs worse than the *perspective, unmarked*. The reason is that, upon hearing a new term agents have to adopt it using a particular landmark. This decision is essentially random, because agents were given no other means of deciding. The *landmark, unmarked* population has more landmarks to choose from. Therefore, agents are more likely to choose wrong. This makes clear that intermediary stages are necessary and that

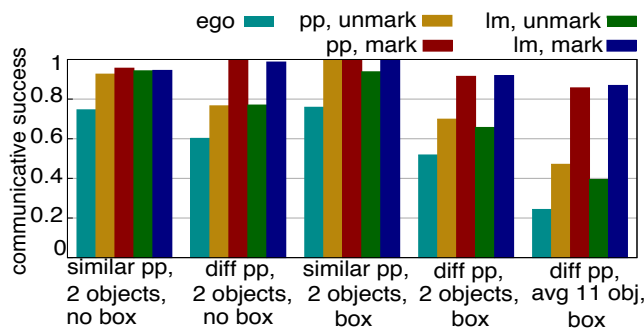


Figure 7: Comparison of different populations forming spatial relation systems (same experimental conditions as in Figure 2, 8 trials). Every population consists of 10 agents which first develop a spatial language system (5000 games) and then their communicative success is measured (2000 interactions).

perspective strategies are suitable intermediaries.

### Evolving Spatial Relations and Landmark Lexicons

Lastly, we turn our attention to whether a lexicon of landmark terms and spatial relations can form simultaneously. Figure 8 shows results for an experiment where agents initially do not have words for landmarks nor do they have spatial relations. The Figure shows that agents can evolve both at the same time.

The following is an example utterance one agent in such a population uttered after 5000 interactions. The agent is using a category that he developed which behaves similarly to “front” and a new word for denoting the box.

- (4) *waketo tabeta*  
category-3 box  
‘to the front of the box’.

### Discussion

The results in this paper suggest two things. First, landmark systems provide a clear communicative advantage over non-landmark systems (shown for projective categories). This means that there is a selective pressure for developing landmark systems for spatial relations such as projective. Second, landmark systems can be culturally negotiated in a population. In particular, we provided simple invention, adoption and alignment operators sufficient for in silico evolution of landmark systems in populations of agents.

The results hint at possible intermediate evolutionary stages. For instance, unmarked cases always perform bet-

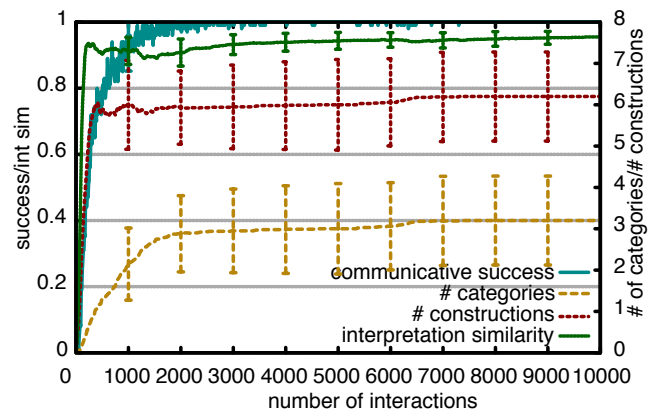


Figure 8: These populations form a system of projective spatial relations and words for marking perspective and landmarks. There is a difference between the average number of words (*# constructions*) and the average number of spatial relations (*# categories*). These are the words denoting landmarks or perspective.

ter than egocentric but worse than marked cases. This is an interesting result before something can be named it has to be conceptualised. In other words, before something can be named as a landmark it needs to have the potential to become a landmark. The unmarked systems therefore provide a necessary stepping stone for marked systems. Another similar example can be found comparing perspective-only to full landmark systems. Using perspective is always less successful than using everything as landmarks. This hints at possible evolutionary trajectories from animate to inanimate landmarks.

The spatial languages evolved in this paper are more or less developing in a vacuum. In the real world, different syntactic and semantic systems and strategies within a language interact and influence each other (Levinson and Wilkins, 2006, provide overviews of different strategies). How to understand and model these interactions remains an open issue although there is some initial work in this direction for colour (Bleys, 2010) and space (Spranger, 2011).

### Acknowledgements

This work was supported by Sony CSL Tokyo and Sony CSL Paris. I am grateful to Masahiro Fujita, Hideki Shimomura and the team at Sony Corporation for making the Sony humanoid robots available for the research discussed in this paper. I thank Luc Steels for help with the robot setup.

### References

- Bleys, J. (2010). Coordination of language strategies based on communicative fitness. In Smith, A. D. M., Schouwstra, M.,

- de Boer, B., and Smith, K., editors, *The Evolution of Language: Evolang 8*, pages 367–368, Singapore. World Scientific.
- Evans, N. and Levinson, S. C. (2009). The myth of language universals: Language diversity and its importance for cognitive science. *Behavioral and Brain Sciences*, 32(05):429–448.
- Herskovits, A. (1986). *Language and spatial cognition*. Studies in Natural Language Processing. Cambridge University Press.
- Levinson, S. C. (2003). *Space in Language and Cognition: Explorations in Cognitive Diversity*. Number 5 in Language, Culture and Cognition. Cambridge University Press.
- Levinson, S. C. and Wilkins, D. (2006). *Grammars of Space*. Cambridge University Press.
- Rosch, E. (1975). Cognitive representations of semantic categories. *Journal of Experimental Psychology: General*, 140:192–233.
- Schulz, R., Stockwell, P., Wakabayashi, M., and Wiles, J. (2006). Towards a spatial language for mobile robots. In Cangelosi, A., Smith, A. D. M., and Smith, K., editors, *The Evolution of Language (Evolang 6)*, pages 291–298, Singapore. World Scientific.
- Spranger, M. (2011). Recruitment, Selection and Alignment of Spatial Language Strategies. In Lenaerts, T., Giacobini, M., Bersini, H., Bourguine, P., Dorigo, M., and Doursat, R., editors, *Advances in Artificial Life, ECAL 2011: Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*, pages 771–778. MIT Press.
- Spranger, M. (2012). The co-evolution of basic spatial terms and categories. In Steels, L., editor, *Experiments in Cultural Language Evolution*, pages 111–141. John Benjamins.
- Spranger, M. and Steels, L. (2012). Emergent Functional Grammar for Space. In Steels, L., editor, *Experiments in Cultural Language Evolution*, pages 207–232. John Benjamins.
- Steels, L. (1995). A self-organizing spatial vocabulary. *Artificial Life*, 2(3):319–332.
- Steels, L. (2012). Evolutionary language games as a paradigm for integrated ai research. In *2012 AAAI Spring Symposium Series, Designing Intelligent Robots*.
- Steels, L. and Belpaeme, T. (2005). Coordinating perceptually grounded categories through language: A case study for colour. *Behavioral and Brain Sciences*, 28:469–529.
- Steels, L. and Loetzsch, M. (2008). Perspective alignment in spatial language. In Coventry, K., Tenbrink, T., and Bateman, J. A., editors, *Spatial Language and Dialogue*, pages 70–88. Oxford University Press.
- Svorou, S. (1994). *The Grammar of Space*, volume 25 of *Typological Studies in Language*. John Benjamins.
- Tinbergen, N. (1963). On aims and methods of ethology. *Zeitschrift für Tierpsychologie*, 20(4):410–433.
- Tomasello, M. (1995). Joint attention as social cognition. In Moore, C. and Dunham, P. J., editors, *Joint Attention: Its Origins and Role in Development*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- van Trijp, R. (2008). The emergence of semantic roles in fluid construction grammar. In Smith, A. and Smith, K. and Ferrer i Cancho, R., editors, *The Evolution of Language (Evolang 7)*, pages 346–353, Singapore. World Scientific Publishing.

# The Artificial Immune Systems Domain: Identifying Progress and Main Contributors Using Publication and Co-Authorship Analyses

Alaa ABI HAIDAR<sup>1,2,3,4</sup>, Adrien SIX<sup>1,2</sup>, Veronique THOMAS-VASLIN<sup>1,2</sup>, and  
Jean-Gabriel GANASCIA<sup>3,4</sup>

<sup>1</sup> CNRS, UMR 7211, Immunology, Immunopathology, Immunotherapy, 75013 Paris

<sup>2</sup> UPMC Univ Paris 06, UMR 7211, Integrative Immunology team, 75013 Paris

<sup>3</sup> CNRS, UMR 7606, ACASA, LIP6, 4 place Jussieu, 75005 Paris

<sup>4</sup> UPMC Univ Paris 06, UMR 7606, ACASA . LIP6 . 4 place Jussieu, 75005 Paris

{alaa.abi\_haidar, adrien.six, veronique.thomas-vaslin, jean-gabriel.

ganascia}@upmc.fr

<http://www.immunocomplexit.net>

**Abstract.** Much can be learned about the progress, fathers and future of a scientific domain from the analysis of a collection of relevant articles and their corresponding authors. Here, we study the highly interdisciplinary domain of Artificial Immune System (AIS) since its birth, a couple of decades ago. We apply Social Network Analysis to the co-authorship network of the most comprehensive publicly accessible AIS bibliography. We automatically extract publication dates and author names from the bibliography and evaluate authors with the highest degree (unique collaborations) and centrality (influence).

Our results highlight the relative growth of publication volume and identify significant contributors in the AIS field. Furthermore, our findings are not only encouraging for the AIS community but may be useful for analyses of other scientific communities and leading contributors therein.

**Keywords:** Artificial Immune Systems, Social Network Analysis, Co-authorship, Information Extraction, Text Mining

## 1 Introduction

Artificial Immune Systems (AIS) are adaptive systems, inspired by theories and observed principles of the immune system, and applied towards solving computational problems [1]. Common AIS techniques are based on specific theoretical models explaining the behavior of the vertebrate adaptive immune system such as negative selection, clonal selection, immune networks and dendritic cells [2]. AIS are mainly classified into two categories: The first one aims at mathematically modeling the immune system to better understand its behavior. The second one uses the immune system as a metaphorical inspiration to engineer algorithms that are capable of learning and solving a huge variety of machine learning problems such as classification, clustering and regression analysis.



AIS is a relatively new field which began in the mid 80's with the modeling and refinement of Jerne's immune network theory [3] by Farmer et al. [4], and later by Bersini and Varela [5, 6]. However, it wasn't until negative selection was used by Forrest [7] for protecting computer networks from viruses that the AIS domain was established. Cooke and Hunt [8, 9] adapted immune networks for classification and Timmis [10] further improved it while De Castro et al [11, 12] worked on aiNet for multimodal function optimization and data analysis. The first book on AIS was edited by Dasgupta in 1998 [13].

In the past few years, several review papers have discussed the slow advances in AIS and proposed improvement strategies through novel and simpler AIS models (inspired by the vertebrate innate immune systems and immune systems of plants) as well as the development of a unified architecture for integration of existing models [14, 2, 15, 16]. Timmis argues that AIS has reached an impasse [2] and Timmis et al pointed out a dearth of theory to justify the use and continuity of AIS [17].

The domain of AIS has existed for a couple of decades but has never been analyzed quantitatively. Furthermore, the fear that the domain is stagnating was only based on qualitative studies targeting specific AIS models and frameworks, such as negative selection, clonal selection and more recently, the danger theory.

In this paper we use techniques from co-authorship network analysis and statistical methods in order to investigate the current state and future of the domain of AIS. Moreover, we identify leading contributors to the field using co-authorship network analysis and discuss our results. In the following section, we discuss the methods used for information extraction and statistical analysis, in addition to social network analysis of the co-authorship network. In section 3, we illustrate and discuss the growth of the AIS domain and compare it to the general growth of Medline indexed articles in general. Finally, in section 3.3, we discuss our results and the future directions of the field.

## 2 Methods

We adopt data mining techniques in order to extract author names and publication dates from the most comprehensive AIS bibliography [18] comprising 1044 articles and 994 authors. We only had access to a binary PDF format of the bibliography and therefore we had to convert it into an ASCII textual format. The converted text was far from the desired clean structured text and therefore a significant amount of manual curation was involved in order to systematically mark the end of the authors' list before automating the extraction of author names.

Publication dates are limited to the year of publication consisting of a 4-digit number  $YYYY$  where  $1980 < YYYY < 2010$  in order to avoid confusion with other numbers such as digital object identifiers (doi), volume numbers and serial numbers. More challenging was the task of author name extraction with some authors having their first names abbreviated others having full first names and others not respecting the first-name-last-name order. We manually restructured

the bibliography, ensuring that last names were preceded by abbreviated or full names. We prepended shared last names with the abbreviated first names to avoid the risk of agglomerating authors with a common last name. For example “X Lee” and “Y Lee” are considered two distinct authors and represented as: X-Lee and Y-Lee, respectively. Still, we face the possibility that two different authors would be counted as one if they share both lastnames and firstnames (at least initial). However, that is outside the scope of our analysis.

## 2.1 Social Network Analysis

Social Network Analysis (SNA) has played major roles in many disciplines in the past few years [19,20]. Co-authorship Networks (CN) are social networks consisting of scientific collaborations and collaborators [21]. In CN, the authors are represented as nodes (or vertices) and collaborations as undirected edges. CN are similar to citation networks in the scientific literature [22]. However, CN have better social and collaborative implications [23]. CN Analysis has already been studied and applied to a couple of fields but never to AIS [23–25]. Indeed, studies on co-authorship analysis have shown how visible and influential can article be [26]. Other studies have focused on examining academic research performance based on a co-authorship network of centrality and gender [27].

We use several existing methods to analyze a CN of AIS as follows:

We used the R package [28] to calculate the degree, closeness, and betweenness centrality for the binary undirected co-authorship network. In the following sections, we illustrate and discuss the 20 highest ranking authors for each of the following metrics:

**Degree** is a measure of the unique number of collaborators an author has.

**Closeness**, that is only applied to the largest (connected) component, is a measure of how authors are directly connected to a well-connected author.

**Betweenness** is a measure of a node’s influence for information flow in the network. Betweenness measures how many times a node is visited when two random nodes are connected through a path of nodes.

For more information about centrality measures and SNA, please refer to [20].

## 2.2 PageRank

PageRank [29], or eigenvector centrality, is used by the Google search engine to determine a page’s relevance or importance. Important pages receive a higher PageRank and are more likely to appear at the top of the search results. PageRank is based on backlinks. The more quality backlinks the higher google pagerank.

Liu et al [23] have applied PageRank to a co-authorship network of the Digital Science domain in order to identify prestigious authors. They transformed each unidirectional edge into a set of bi-directional, symmetrical edges.

### 2.3 AuthorRank

Liu et al [23] also define a modification of PageRank, that they call AuthorRank. PageRank assumes that when a node  $A$  connects to  $n$  other nodes, it receives a weight based on an equal fraction  $\frac{1}{n}$ , whereas AuthorRank attributes different weights for each author based on the number of their publications in common *ibid*.

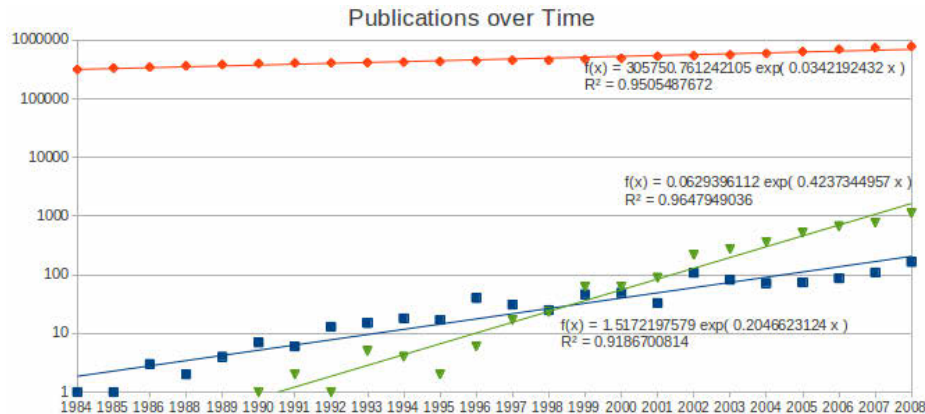
Both PageRank and AuthorRank are measures of prestige that we use to identify leading authors in the domain of AIS as discussed in the following section.

We used the R package [28] to implement PageRank and AuthorRank to rank the top ranking authors and to visualize the co-authorship network using a Fruchterman Reingold Layout.

## 3 Results and Discussion

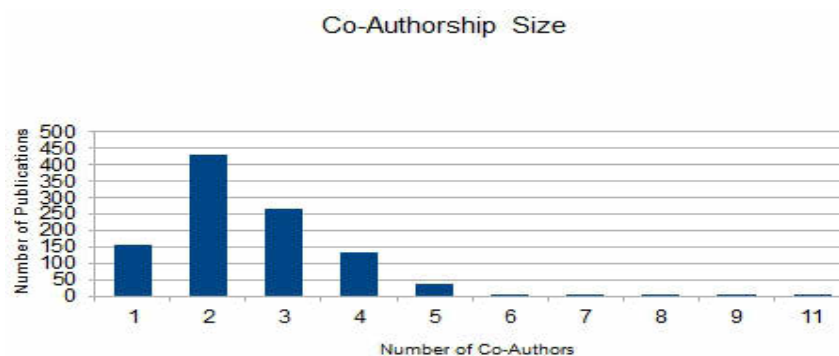
### 3.1 Publication Distribution Analysis

In order to answer the numerous doubts about a decline in the AIS field, we have measured the number of publications that are relevant to the field over time. Furthermore, we have fitted our observation using exponential functions to study the growth of publication size over time and to predict it over the years to come. We have used the coefficient of determination  $R^2$  in order to validate the fitness function and its prediction as shown in figure 1.



**Fig. 1.** Number of Publications per year 1) according to the AIS bibliography [blue], 2) according to google scholar results for “Artificial Immune Systems” [green] and 3) for all indexed Pubmed articles [red] per year. The results show a relative growth in the domain based on both AIS bibliography and Google Scholar results. The exponential fits are validated using the  $R^2$  coefficient.

Contrary to previous fears of a stagnating AIS field, we have shown that the field of AIS is ever growing by measuring the number of publications over time. Indeed, we have used exponential fitness functions in the form of  $f(x) = c.e^{(x.b)}$ , where  $c$  is a constant,  $x$  is the year index and  $b$  is the growth factor. We have compared the growth in the volume of AIS articles according to our AIS bibliography ( $b = 0.2$ ) to that of indexed Pubmed articles ( $b = 0.03$ ) between the years of 1984 and 2008<sup>1</sup> using Corlan's Medline trend (<http://dan.corlan.net/medline-trend.html>) to cement our conclusion regarding the relative expansion of the AIS domain. Furthermore, we add to our perspective the number of articles returned by Google Scholar for the query "Artificial Immune Systems" which results in a faster growing trend ( $b = 0.4$ ) as shown in green in figure 1. Dasgupta's Bibliography includes conference proceedings, as can be found in Google Scholar, whereas PubMed indexes journal articles only. However, we argue that in the fields of informatics and engineering, conference and workshop publications typically have higher impact



**Fig. 2.** The distribution of publications per authors forms a gaussian distribution around two authors.

In addition, we have analyzed the number of authors per publication that is an indication of collaboration strength. As shown in figure 2, most collaborations include two authors. Moreover, there are more 3 co-authored publications than (and almost as many 4 co-authored publications as) single authored ones. We presume that this may be as a result of the field of AIS being a very collaborative one. Indeed, a highly interdisciplinary field such as AIS invites collaborations from the fields of immunology, systems biology, artificial intelligence, machine learning, data mining...etc. Similar studies have been conducted on the field of the Digital Library Research Community [23] yielding similar results with 28.5% of papers authored by 2 authors, 23.6% by 3 authors, 19.6% by a single author and 9.4% by 4 authors.

<sup>1</sup> The years covered in the AIS Bibliography



### 3.2 Co-Authorship Network Analysis

The co-authorship network is summarized in table 1 and visualized in figure 3.

| Network Statistics     |        |
|------------------------|--------|
| Articles               | 1044   |
| Number of Vertices     | 994    |
| Number of Edges        | 1768   |
| Network Density        | 0.0036 |
| Number of Clusters     | 159    |
| Largest Component Size | 55%    |

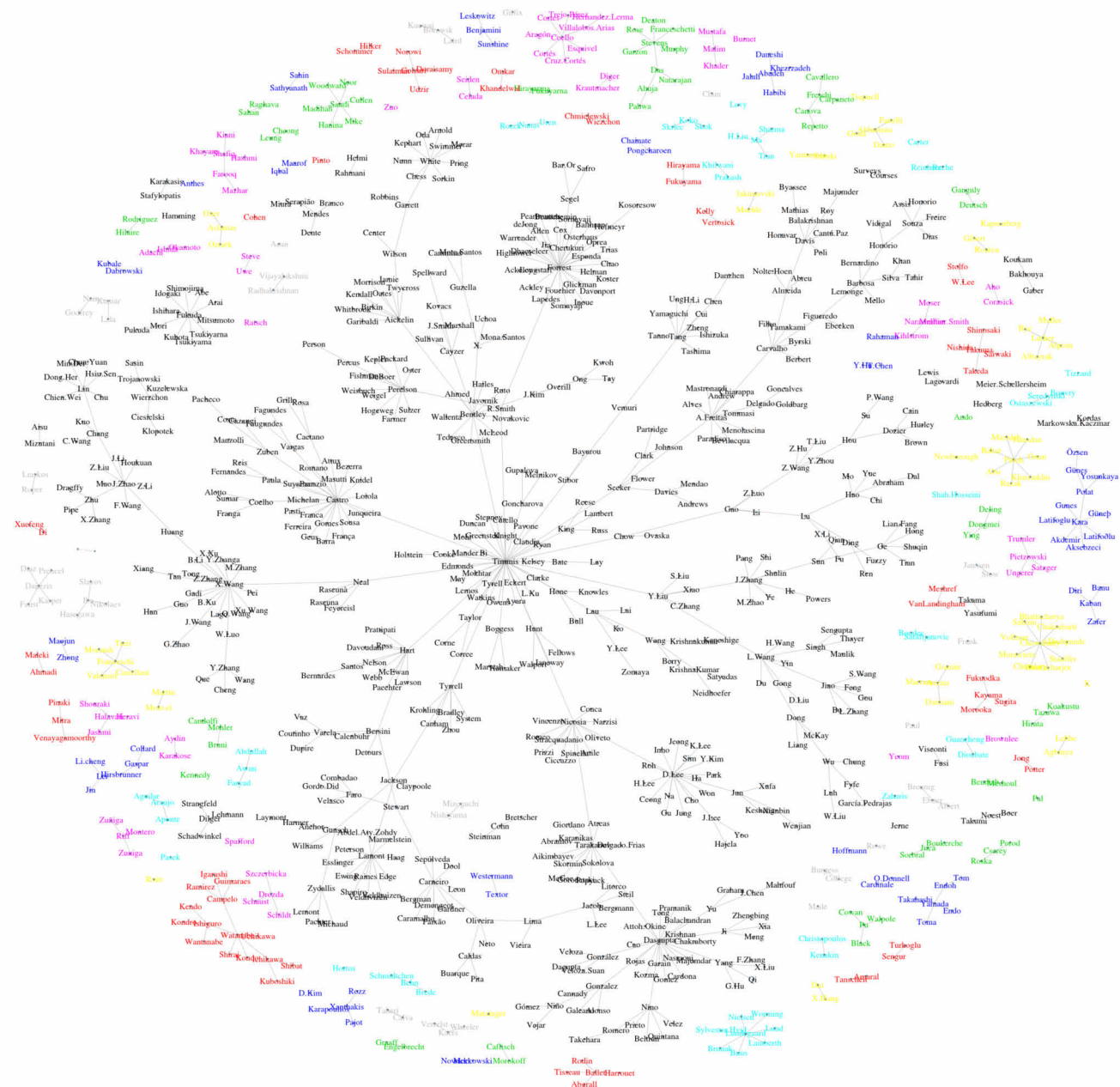
**Table 1.** Co-Authorship network summary

The number of publications per author may be a good indicator of an author's contribution to the field however that can be biased in favor of non-collaborative authors with many published articles. The author's degree, or number of unique collaborators, can be a better measure of an author's collaborative efforts in a field. Degree centrality measures authors' connectivity with immediate neighbors or collaborators. Some authors may, however, be locally well connected but not globally with the entire network. Closeness centrality expands on degree centrality and favors authors that are connected (directly or indirectly) to as many authors in the network. Betweenness is another measure of centrality that measures how often a node is on a shortest path between any two random nodes in the network. Betweenness conveys the role of an author as an information spreader or a hub. The authors with the highest number of publications, degree, betweenness and closeness are listed in table 2. Authors with the highest degree, betweenness and closeness measures are illustrated in figure 4.

Several studies have analysed Co-authorship Network (CN) components for various scientific domains. Nascimento [24] reports the largest component in SIGMODs<sup>2</sup> CN having about 60% of all authors. Newman [25] has studied several CN with the smallest "largest component", containing 57.2% of all authors. Liu et al [23] report in the JCDL<sup>3</sup> CN the largest component of only 38% (599 authors) of all authors for the years between 1994 and 2003. The low percentage may be due to a relative immaturity of the Digital Library (DL) field. The AIS CN has the largest component of 55% (550 authors) of all authors, thus showing a relative maturity of the field. We suspect that the maturity of AIS is related to 10 years of the dedicated conference, ICARIS, since 2002 and 20 years since the beginning of AIS, i.e. twice as longer than the DL field. The largest component of 550 authors is visualized in black at the center of figure 3, whereas the remaining components alternate in various colors around it. Moreover, table 3

<sup>2</sup> SIGMOD is a Special Interest Group on Management Of Data under the Association of Machinery, ACM

<sup>3</sup> Joint Conference on Digital Library



**Fig. 3.** A visualization of the AIS co-authorship network with nodes representing authors and edges representing collaborations (with at least one co-authored article). Each component is represented in a different color. In particular, the largest component (in black) contains 55% of all authors.

| Rank | Author     | Publications | Author    | Degree | Author    | Betweenness | Author     | Closeness |
|------|------------|--------------|-----------|--------|-----------|-------------|------------|-----------|
| 1    | Timmis     | 83           | Timmis    | 50     | Timmis    | 86260       | Timmis     | 0.22405   |
| 2    | Dasgupta   | 71           | Forrest   | 35     | Dasgupta  | 35531       | R-Smith    | 0.22391   |
| 3    | Forrest    | 68           | Castro    | 29     | Forrest   | 20535       | Bentley    | 0.22390   |
| 4    | Castro     | 58           | Dasgupta  | 28     | Castro    | 18859       | Nicosia    | 0.22389   |
| 5    | Zuben      | 42           | Perelson  | 28     | X-Wang    | 17737       | Tyrrell    | 0.22389   |
| 6    | Aickelin   | 38           | X-Wang    | 24     | A-Freitas | 17628       | Chow       | 0.22386   |
| 7    | Perelson   | 35           | D-Lee     | 22     | Chow      | 16984       | Y-Liu      | 0.22386   |
| 8    | Hart       | 20           | Zuben     | 20     | Hart      | 16183       | Clark      | 0.22385   |
| 9    | Gonzalez   | 19           | J-Kim     | 18     | R-Smith   | 15714       | Neal       | 0.22385   |
| 10   | Stibor     | 18           | Lamont    | 18     | Tyrrell   | 15451       | Lau        | 0.22384   |
| 11   | Neal       | 17           | Aickelin  | 17     | Y-Liu     | 14936       | Forrest    | 0.22384   |
| 12   | Hunt       | 17           | Tarakanov | 16     | Bentley   | 14827       | Cutello    | 0.22384   |
| 13   | Bersini    | 17           | A-Freitas | 15     | Perelson  | 13960       | Pavone     | 0.22384   |
| 14   | Lamont     | 16           | Nicosia   | 15     | Stewart   | 13312       | Goncharova | 0.22384   |
| 15   | Lau        | 16           | Faro      | 15     | Aickelin  | 12671       | Knight     | 0.22384   |
| 16   | Esponda    | 15           | Bentley   | 13     | Gao       | 12462       | Stepney    | 0.22384   |
| 17   | J-Kim      | 15           | Oliveira  | 13     | Carvalho  | 11618       | Dasgupta   | 0.22383   |
| 18   | Bentley    | 15           | Hart      | 13     | Jackson   | 11594       | Castro     | 0.22383   |
| 19   | Greensmith | 15           | Clark     | 12     | Huang     | 11481       | Hart       | 0.22383   |
| 20   | Tarakanov  | 15           | Gonzalez  | 12     | Bersini   | 11119       | A-Freitas  | 0.22382   |

**Table 2.** AIS authors ranked according to their number of publications, degree (or number of unique collaborators), betweenness and closeness.

| Component Size       | 1  | 2  | 3  | 4  | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 15 | <b>550</b> |
|----------------------|----|----|----|----|---|---|---|---|---|----|----|----|----|------------|
| Number of Components | 39 | 60 | 22 | 15 | 9 | 4 | 1 | 2 | 1 | 2  | 1  | 1  | 1  | <b>1</b>   |

**Table 3.** AIS Co-authorship Network component sizes and frequencies with the largest component boldened

lists the component sizes and frequencies of the AIS CN showing a significant amount of smaller components. This is suggestive of many existing AIS sub-communities, that collaborate individually but can eventually collaborate with other sub-communities .

Table 2 distinguishes Jon Timmis as a top contributor and collaborator in this field, however, we are also interested in identifying other contributors in the AIS domain like the founding fathers such as Forrest and De Castro. In table 2, we identify authors with significantly high centrality (top 20 in degree and either betweenness or closeness<sup>4</sup>) such as Nicosia, A.Freitas in the AIS CN regardless of their relatively lower number of publications (not in the top 20 in number of publications). Conversely, authors such as Neal and Stibor rank amongst the top 20 for their number of publications but do not rank as highly

<sup>4</sup> We look at the betweenness and closeness together given the minute variation in the top ranked authors according to closeness centrality shown in table 2

with the centrality measures. We assume this is due to their collaboration with a fewer selection of authors. Nevertheless, Neal and Stibor reappear in the top ranking authors according to AuthorRank in table 4 for their collaborations with few but “prestigious” authors (in terms of AuthorRank) such as Timmis, Hunt and Eckert.

PageRank and AuthorRank are measures of prestige. An author ranks highly if he or she collaborate with many authors that, in turn, have many collaborators – *ad libitum*. AuthorRank gives more weight to repeated collaborations (in a weighted network), whereas PageRank distributes weights evenly (in a boolean network). The top 20 authors according to PageRank and AuthorRank are listed in table 4.

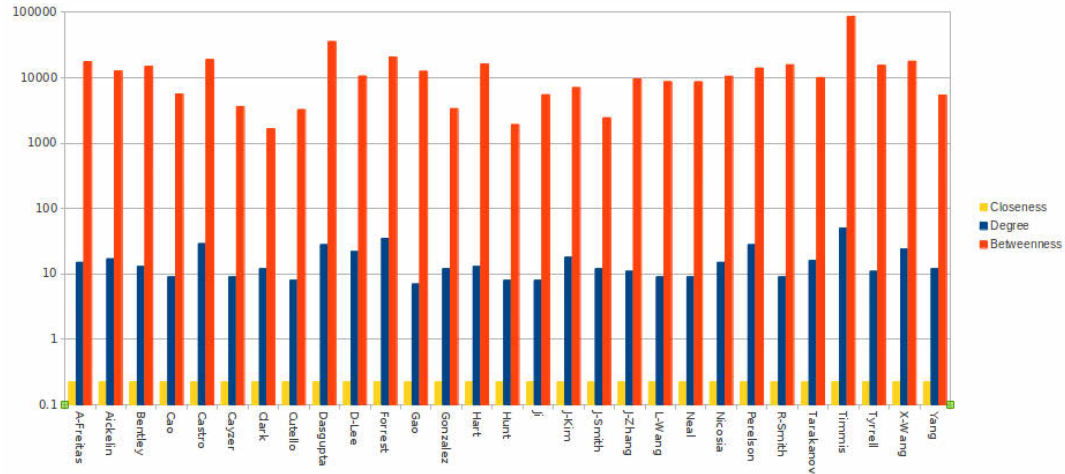
| Rank | PageRank  | AuthorRank |
|------|-----------|------------|
| 1    | Timmis    | Timmis     |
| 2    | Forrest   | Castro     |
| 3    | Dasgupta  | Neal       |
| 4    | Perelson  | Stibor     |
| 5    | Castro    | Hunt       |
| 6    | X-Wang    | Knight     |
| 7    | D-Lee     | Eckert     |
| 8    | Zuben     | Zuben      |
| 9    | Lamont    | A-Freitas  |
| 10   | Aickelin  | Watkins    |
| 11   | J-Kim     | Andrews    |
| 12   | Coello    | Clark      |
| 13   | Tarakanov | Hart       |
| 14   | Hart      | Ayara      |
| 15   | Nicosia   | Lemos      |
| 16   | Chowdhury | Mohr       |
| 17   | Fukuda    | Secker     |
| 18   | Yang      | Owens      |
| 19   | Gonzalez  | Hone       |
| 20   | J-Zhang   | Kelsey     |

**Table 4.** Authors ranked according to the metrics of PageRank and AuthorRank.

We are equally interested in identifying contributors to the field with leading AIS theoretical models, such as Jerne, Matzinger, Perelson, Bersini, Varela, Carneiro and Greensmith, recurrent committee members of the International Conference of Artificial Immune Systems (ICARIS) such as Timmis, Forrest and Nicosia, as well as principle investigators or team leaders in AIS, such as Hart, Timmis, Von Zuben and D. Dasgupta. Many of these names appear repeatedly in table 2 of top ranking authors according to centrality measures, and in table 4 of top ranking authors according to PageRank and AuthorRank.



Finally, we are interested in understanding large network components that are disconnected from the largest component such as those led by Fukuda, Kara, Watanabe and Coello. We presume that separation is due to language and geographic barriers but we hope to have a more integrated network or more methodological explanations about this segregation in the near future. We are as well interested in understanding cluster properties of the largest component mainly led by team leaders, namely, Forrest, Timmis, Hart, Dasgupta and Tarakanov.



**Fig. 4.** Alphabetically sorted list of authors with the highest degree, betweenness and closeness

### 3.3 Conclusion and Future Directions

Several reviews have discussed advances in the field of Artificial Immune Systems but all from a qualitative perspective. Recent reviews have alluded to a stagnation in the field of AIS. In this work, we investigated these questions from a quantitative perspective. Our results have shown that the field has been growing ever since it was established for the past couple of decades. In addition, we have identified leading contributors by co-authorship network analysis based on AIS-relevant publications.

We acknowledge that the bibliography may be biased towards an engineering perspective as it is maintained by a computer scientist thus dismissing fundamental contribution from the theoretical immunology perspective. However, our method can be applied to any bibliography (structured or unstructured) in any scientific domain. Hence, we expect our analytical method not only to motivate the AIS community and encourage external scientists to entertain the challenges presented by AIS, but also to be a benchmark for scientific domain analyses.

**Acknowledgments.** This research has been supported by the DIM ISC 2011 Grant "Problématiques transversales aux systèmes complexes" Region Paris Ile-de-France through the Institute of Complex Systems (ISC-PIF). We acknowledge the help of Hugues RIPOCHE with the extraction of dates and the insights brought by Luis M. ROCHA and Chris MCEWAN.

## References

1. De Castro, L., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer (2002)
2. Timmis, J.: Artificial immune systems today and tomorrow. *Natural Computing* **6**(1) (2007) 1–18
3. Jerne, N.: Towards a network theory of the immune system. *Ann Immunol (Paris)* **125**(1-2) (1974) 373–89
4. Farmer, J., Packard, N., Perelson, A.: The immune system, adaptation, and machine learning. *Physica* **22**(2) (1986) 187–204
5. Varela, F., Coutinho, A.: Second generation immune networks. *Immunol Today* **12**(5) (1991) 159–66
6. Bersini, H., Varela, F.: Hints for adaptive problem solving gleaned from immune networks. *Parallel problem solving from nature* (1991) 343–354
7. Forrest, S., Perelson, A., Allen, L., Cherukuri, R.: Self-nonsel self discrimination in a computer. *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy* **212** (1994)
8. Cooke, D., Hunt, J.: Recognising promoter sequences using an artificial immune system. *Proc Int Conf Intell Syst Mol Biol* **3** (1995) 89–97
9. Hunt, J., Cooke, D.: Learning using an artificial immune system. *Journal of Network and Computer Applications* **19**(2) (1996) 189–212
10. Timmis, J., Neal, M.: A resource limited artificial immune system for data analysis. *Knowledge-Based Systems* **14**(3-4) (2001) 121–130
11. de Castro, L., Timmis, J.: An artificial immune network for multimodal function optimization. *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on* **1** (2002)
12. de Castro, L., Von Zuben, F.: aiNet: An Artificial Immune Network for Data Analysis. *Data Mining: A Heuristic Approach* (2002)
13. Dasgupta, D.: Artificial Immune Systems and Their Applications. Springer-Verlag New York, Inc. Secaucus, NJ, USA (1998)
14. Dasgupta, D.: Advances in artificial immune systems. *Computational Intelligence Magazine, IEEE* **1**(4) (2006) 40–49
15. Twycross, J., Aickelin, U.: Biological Inspiration for Artificial Immune Systems. Springer Verlag, LNCS **4628** (2007) 300
16. Hart, E., Timmis, J.: Application areas of AIS: The past, the present and the future. *Applied Soft Computing Journal* **8**(1) (2008) 191–201
17. Timmis, J., Hone, A., Stibor, T., Clark, E.: Theoretical advances in artificial immune systems. *Theoretical Computer Science* **403**(1) (2008) 11–32
18. Dasgupta, D., Majumdar, N., Nino, F.: Artificial immune systems: A bibliography. Computer Science Division, University of Memphis, Technical Report (2007)
19. Barabasi, A., Crandall, R.: Linked: The new science of networks. *American journal of Physics* **71** (2003) 409

20. Wassermann, S., Faust, K.: Social network analysis: Methods and applications. New York (1994)
21. Ferrara, A., Salini, S.: Ten challenges in modeling bibliographic data for bibliometric analysis. *Scientometrics* 1–21 10.1007/s11192-012-0810-x.
22. Garfield, E.e.a.: Citation analysis as a tool in journal evaluation. (1972)
23. Liu, X., Bollen, J., Nelson, M., Van de Sompel, H.: Co-authorship networks in the digital library research community. *Information processing & management* **41**(6) (2005) 1462–1480
24. Nascimento, M., Sander, J., Pound, J.: Analysis of sigmod’s co-authorship graph. *ACM SIGMOD Record* **32**(3) (2003) 8–10
25. Newman, M.: Scientific collaboration networks. i. network construction and fundamental results. *Physical review E* **64**(1) (2001) 016131
26. Egghe, L., Guns, R., Rousseau, R.: Measuring co-authors contribution to an articles visibility. *Scientometrics* 1–13 10.1007/s11192-012-0832-4.
27. Badar, K., Hite, J., Badir, Y.: Examining the relationship of co-authorship network centrality and gender on academic research performance: the case of chemistry researchers in pakistan. *Scientometrics* 1–21 10.1007/s11192-012-0764-z.
28. Team, R., et al.: R: A language and environment for statistical computing. R Foundation for Statistical Computing Vienna Austria (01/19) (2010)
29. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. (1999)

# The Transmission of Migratory Behaviors

Olaf Witkowski and Geoff Nitschke<sup>1</sup>

Ikegami Laboratory, Interdisciplinary Studies Department, University of Tokyo, Japan

olaf@sacral.c.u-tokyo.ac.jp

<sup>1</sup>Department of Computer Science, University of Cape Town, South Africa

gnitschke@cs.uct.ac.za

## Extended Abstract

In nature, animals rely upon migratory behaviors in order to adapt to seasonal variations in their environment. However, the transmission of migratory behaviors within populations (either during lifetimes or throughout successive generations) is not well understood (Bauer et al., 2011). In *Artificial Life* research, *Agent Based Modeling* (ABM) is a bottom-up approach to study evolutionary conditions under which adaptive group behavior emerges. ABM is characterized by synthetic methods (understanding via building), and is becoming increasingly popular in animal behavior research (Sumida et al., 1990). Combining an *Artificial Neural Network* (ANN) and *Evolutionary Algorithm* (EA) for adapting agent behavior (Yao, 1993) has received significant research attention (Phelps and Ryan, 2001), (Lee, 2003).

ABM is an analogical system that aids ethologists in constructing novel hypotheses, and allow the investigation of emergent phenomena in experiments that could not be conducted in nature (Webb, 2009). Numerous studies in ethology have formalized mathematical models of migratory patterns in various species (Bauer et al., 2011). However, there have been few studies that examine ontological and phylogenetic conditions requisite for emergent migratory behavior. ABM is advantageous (compared to formal mathematical models of migratory behavior), since various evolutionary processes can be simulated, and variations in resultant migratory behaviors examined. For example, ABM has been used to predict the consequences of forced human migrations (Edwards, 2009), and migratory behavior between groups of Macaque monkeys (Hemelrijk, 2004).

In this research, ABM is used to investigate a hypothesis posited in ethological literature: that migratory behavior is adopted as an adaptive foraging behavior, where such behavior is either genetically or culturally determined (Huse and Giske, 1998). This study aims to investigate the evolutionary and cultural conditions that give rise to migratory behaviors and thus adaptive foraging. In cultural behavioral transmission, ontogenetic transfer occurs between agents during their lifetime. Alternatively, migratory behavior is phylogenetically transmitted through successive generations (Bauer

et al., 2011). A minimalist simulation model (distribution of four food patches and 200 agents on a grid) demonstrates the impact of ontogenetic versus phylogenetic transmission of migratory behavior and thus agent group adaptivity.

Agents use an ANN controller (figure 1, left). ANN connection weights are adapted with an EA. Agent fitness is the food amount consumed during a lifetime (200 iterations). The EA selects for effective foraging behaviors, which depends upon agents periodically migrating to where food is plentiful. Stimuli for migratory behavior take the form of cyclic *seasons* in the environment and agents signaling their movement direction to neighbors. When it is *winter* (food is scarce) in one half of the environment, it is *summer* (food is plentiful) in the other half, where each seasonal cycle (50 iterations) the winter and summer zones are switched.

Each iteration, agents receive the sensory inputs: *signal* from the closest agent, their current *fitness* and *recurrent* connections (activation value of the hidden layer in the previous iteration). Agent behavior is: *move* to an adjacent grid square, *mimic* or *mate* with a neighboring agent. The output with the highest activation is selected (figure 1, left). Each iteration, agents also emits a signal (output not depicted in figure 1), conveying the sender's current direction of movement and thus indicating migratory behavior.

Via choosing to *mimic* or *mate*, agents either imitate their neighbor's migratory behaviors or pass genetically encoded migratory behaviors onto their offspring. If an agent mimics, it copies the ANN connection weights of its closest neighbor, thus mimicking its neighbors behavior, which includes the *direction* signal sent each iteration. If an agent mates, fitness proportionate selection (Eiben and Smith, 2003) is used to select a mate from the agent population. Genotypes (floating-point value strings) encoding the ANNs are recombined using 2-point crossover (Eiben and Smith, 2003). Two child ANNs are produced and replace the parents to keep the population size constant. If an agent moves, then it moves one grid cell *north*, *south*, *east*, or *west* (figure 1, left).

Figure 1 (center and right) illustrates agent adaptation occurring over evolutionary time. Agents become effective gatherers via learning a migration behavior allowing them



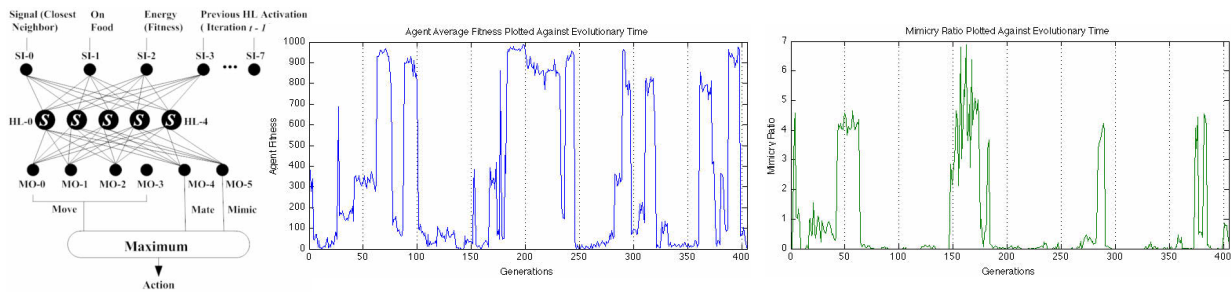


Figure 1: *Left*: Each agent is a recurrent feed-forward ANN. *SI*: Sensory Input. *MO*: Motor Output. *HL*: Hidden Layer. *Center*: Average agent group fitness over 400 generations of neuro-evolution. *Right*: Average mimicry ratio over 400 generations.

to move about the environment in synchronization with the seasons (moving to where food is plentiful). Figure 1 also delineates a cyclic process in agent adaptive behavior, and the relationship between fitness and behavioral mimicry. *Mimicry ratio* indicates the average preference of an agent to mimic over another behavior. Figure 1 (center) also indicates agents periodically adapt to effective foraging behavior (indicated by fitness spikes). Fitness increases result from agents adopting migratory behaviors to adapt to the environment's seasonal variation, where such increases are enhanced by behavioral mimicking in preceding generations.

We hypothesize that subsequent periodic fitness drops, and preceding mimicry ratio decreases (figure 1, right), result from the selection and propagation of fit yet non-robust behaviors. Periodic fitness increases (figure 1, center) indicate the agents converge towards an effective gathering behavior. However, concurrently, behavioral heterogeneity is bred out of the population. Convergence results in a homogenous agent group that is unable to cope with seasonal variation in the environment. This in turn causes the periodic fitness crashes (figure 1, center), where most of the population dies off, and only those agents with robust behaviors (suited to seasonal variation) survive and are selected for.

Thus, behavioral takeover in the population (accelerated by behavioral mimicry and fitness proportionate selection) results in a largely homogenous population with low genotype and fitness diversity (Wineberg and Oppacher, 2003) and non-robust behaviors. Subsequent fitness decreases reintroduce behavioral heterogeneity (and fitness diversity) into the population and allow agents to re-adapt to the environment's seasonal variation via adopting a migratory behavior. Figure 1 (center and right) also indicates that variations in the mimicry rate impact the rate of agent adaptation and re-adaptation, as well as the duration of fitness spikes. That is, fitness increases are correlated with high mimicry ratios and fitness crashes cause behaviors containing the propensity to mimic to be periodically lost, and then rediscovered in the subsequent re-adaptation phase.

Whilst preliminary results indicate the importance of behavioral mimicry and genetic transmission of migratory be-

haviors to a population's overall adaptivity (supporting ethological research), their contribution to adaptive behavior is subject to ongoing research. Current investigation is of conditions under which cultural versus genetic transmission of migratory behaviors prevail, and the impact of lifetime duration on cultural and genetic transmission of behaviors.

## References

- Bauer, S., Nolet, B., Giske, J., Chapman, J., Akesson, S., Hedenstrom, A., and Fryxell, J. (2011). Cues and decision rules in animal migration. In *Animal Migration: A Synthesis*, pages 69–87. Oxford University Press, Oxford, UK.
- Edwards, S. (2009). *The Chaos of Forced Migration: A Means of Modeling Complexity for Humanitarian Ends*. Oxford University Press, Oxford, United Kingdom.
- Eiben, A. and Smith, J. (2003). *Introduction to Evolutionary Computing*. Springer-Verlag, Berlin, Germany.
- Hemelrijk, C. (2004). The use of artificial-life models for the study of social organization. In *Macaque Societies: A Model for the Study of Social Organization*, pages 295–313. Cambridge University Press, Cambridge, UK.
- Huse, G. and Giske, J. (1998). Ecology in mare pentium: an individual-based spatio-temporal model for fish with adapted behaviour. *Fisheries Research*, 37(1):163178.
- Lee, M. (2003). Evolution of behaviors in autonomous robots using artificial neural networks and genetic algorithms. *Information Science*, 155(1):43–60.
- Phelps, S. and Ryan, M. (2001). History influences signal recognition: neural network models of tungara frogs. *Proceedings of the Royal Society B*, 267(1):1633–1639.
- Sumida, B., Houston, A., McNamara, J., and Hamilton, W. (1990). Genetic algorithms and evolution. *Journal of Theoretical Biology*, 147(1):59–84.
- Webb, B. (2009). Animals versus animats: Or why not model the real iguana? *Adaptive Behavior*, 17(4):269–286.
- Wineberg, M. and Oppacher, F. (2003). The underlying similarity of diversity measures used in evolutionary computation. In *Proceedings of the Fifth Genetic and Evolutionary Computation Conference*, pages 1493–1504, Berlin. Springer.
- Yao, X. (1993). Evolutionary artificial neural networks. *International Journal of Neural Systems*, 4(3):203–222.

# Adaptive long-range migration promotes cooperation under tempting conditions

Genki Ichinose<sup>1</sup>, Masaya Saito<sup>1</sup>, Hiroki Sayama<sup>2</sup> and David Sloan Wilson<sup>3</sup>

<sup>1</sup>Anan National College of Technology, 265 Aoki Minobayashi, Anan, Tokushima 774-0017, Japan

<sup>2</sup>Collective Dynamics of Complex Systems Research Group,

Binghamton University, State University of New York, Binghamton, NY 13902-6000, USA

<sup>3</sup>Departments of Biological Sciences and Anthropology,

Binghamton University, State University of New York, Binghamton, NY 13902-6000, USA

ichinose@anan-nct.ac.jp

## Abstract

Migration is a fundamental trait in humans and animals. Recent studies investigated the effect of migration on the evolution of cooperation, showing that contingent migration favors cooperation in spatial structures. In those studies, only local migration to immediate neighbor sites was considered, while long-range migration has not been considered yet, partly because the long-range migration has been generally regarded as harmful for cooperation as it would bring the population to a well-mixed state that favors defection. Here, we studied the effects of adaptive long-range migration on the evolution of cooperation through agent-based simulations of a spatial Prisoner's Dilemma game in which individuals can jump to a farther site if they are surrounded by more defectors. Our results show that adaptive long-range migration strongly promotes cooperation, especially under conditions where the temptation to defect is considerably high. These findings demonstrate the significance of adaptive long-range migration, a naturally observed migration style in human and animal behaviors, for the evolution of cooperation.

Migration is a fundamental trait in humans and animals. Recent studies investigated the effect of migration on the evolution of cooperation, showing that contingent migration favors cooperation in spatial structures. One limitation in the earlier contingent migration studies is that migration was commonly assumed to be limited only to immediate neighbor sites of each individual. To our knowledge of the relevant literature, no existing work addressed the effect of *adaptive long-distance migration* on the evolution of cooperation, partly because the long-range migration has been generally regarded as harmful for cooperation as it would bring the population to a well-mixed state that favors defection. We believe, however, that adaptive long-range migration may be beneficial for cooperators because they may be able to escape from defector-dominated areas at once and potentially land in a community of other cooperators.

Here, we studied the effects of adaptive long-range migration on the evolution of cooperation. For this, we developed an agent-based model in which individuals play the Prisoner's Dilemma (PD) game with local nearest neighbors on a regular spatial grid and may migrate to another site if the local situation is unfavorable. The key unique contribu-

tion of our work is to use an adaptive distance for long-range migration based on the number of defectors in the individual's vicinity, so that the possible distance of the adaptive migration is longer with more defectors in the area nearby. The square lattice is composed of  $L \times L$  sites with periodic boundary conditions. Each site is either empty or occupied by one individual. Empty sites represent spatial regions that individuals can migrate to. Initially, individuals are randomly distributed over the square lattice. One half of them are set to be cooperators while the other defectors. Their population density is given by  $\rho$ . The population density remains constant throughout a simulation run, since individuals will never die or be born.

Individuals are updated asynchronously in a randomized sequential order. The algorithm for updating an individual consists of the following three phases:

1. *Migration.* First, to decide the site to move into, the individual counts the number of defectors,  $n_D$ , within its  $n$  neighbors. Using  $n_D$ , the maximum migration distance is calculated as  $d_{\max} = \text{round}((n_D/n)^\alpha d)$ , where  $\alpha$  and  $d$  are experimental parameters.  $\text{round}(\dots)$  is used here because the space is discrete. Then the individual moves to a randomly selected empty site within a  $(2d_{\max} + 1)$ -by- $(2d_{\max} + 1)$  square region around its original location.
2. *Game play.* After the migration, the individual plays the PD game with other individuals within its  $n$  nearest neighbor sites and accumulates the payoffs resulting from the games. In each game, two individuals decide whether to cooperate or defect simultaneously based on their current strategies. They both obtain payoff  $R$  for mutual cooperation while  $P$  for mutual defection. If one selects cooperation while the other does defection, the former gets the sucker's payoff  $S$  while the latter gets the highest payoff  $T$ , the temptation to defect. The relationship of the four payoffs is usually  $T > R > P > S$  in PD games. Following the parameter settings used in the earlier studies, we used  $P = 0$ ,  $R = 1$ , and  $S = 0$ , while  $T > 1$ , the temptation to defect, was varied as an experimental parameter.

3. *Strategy updating.* Once all the games are over, the individual imitates the strategy of the other individual that achieved the highest total payoff among its neighbors.

When all individuals finish conducting the three phases above, it constitutes one time step of the simulation, which we refer to as a “generation.”

The parameter setting used for the experiments is as follows:  $L = 50$ ,  $\rho = 0.3$ ,  $n = 8$  (Moore neighborhood). We systematically varied values of  $\alpha$  and  $d$  to represent the following six different types of migration:

- $\alpha = 0$ ;  $d = 1$ : random short-range migration (RSM)
- $\alpha = 0$ ;  $d = 8$  ( $> 1$ ): random long-range migration (RLM)
- $\alpha = 0$ ;  $d = 24$  ( $\sim L/2$ ): random global migration (RGM; an extreme case of RLM)
- $\alpha = 1/3, 1$ , or  $3$ ;  $d = 1$ : adaptive short-range migration (ASM)
- $\alpha = 1/3, 1$ , or  $3$ ;  $d = 8$  ( $> 1$ ): adaptive long-range migration (ALM)
- $\alpha = 1/3, 1$ , or  $3$ ;  $d = 24$  ( $\sim L/2$ ): adaptive global migration (AGM; an extreme case of ALM)

The main experimental measurement obtained from each simulation run is the equilibrium fraction of cooperators,  $\bar{f}_c$ , after a certain time period elapses.

By conducting evolutionary simulations, we found that ASM, ALM, and AGM strongly promotes cooperation compared to RSM, RLM, and RGM, especially under conditions where the temptation to defect is considerably high. Figure 1 shows  $\bar{f}_c$  as a function of  $T$ . For the low values of  $T$ , all migration types sustained a high level of cooperation (except for  $\alpha = 3$  that made migration rather infrequent). In highly tempting conditions ( $T > 3$ ), cooperators were easily exploited by defectors in RSM, RLM, and RGM. In ASM, cooperators were allowed to move in response to the presence of defectors, but they often remained near the defectors and were exploited by them. In ALM and AGM, in contrast, cooperators were able to secure a certain distance from defectors after the adaptive migration, and thus cooperation was promoted even under highly tempting conditions. Even complete segregation between cooperators and defectors was possible in ALM (for  $\alpha = 1/3$ ) and AGM (for  $\alpha = 1/3, 1$ ) (Fig. 2).

In conclusion, we investigated the effect of adaptive long-range migration on the evolution of cooperation in a spatial population model. Compared to earlier studies that assumed only short-range migration, we found that adaptive long-range migration strongly enhanced the evolution and maintenance of cooperative behavior, especially under high tempting conditions where other forms of migration did not.

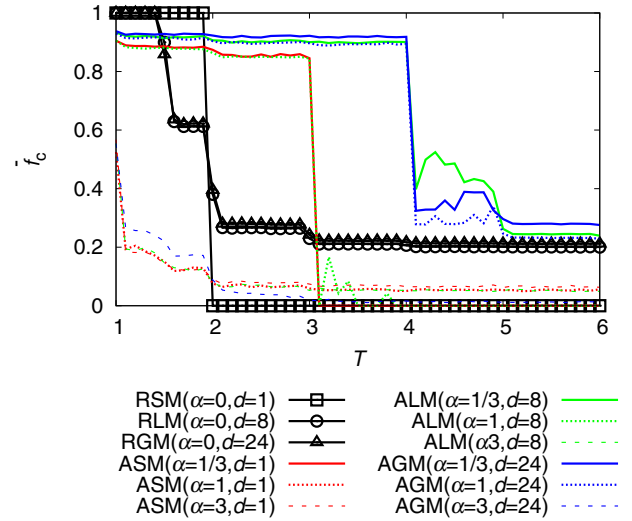


Figure 1: Equilibrium fraction of cooperators ( $\bar{f}_c$ ) as a function of temptation to defect ( $T$ ).  $\rho = 0.3$ .

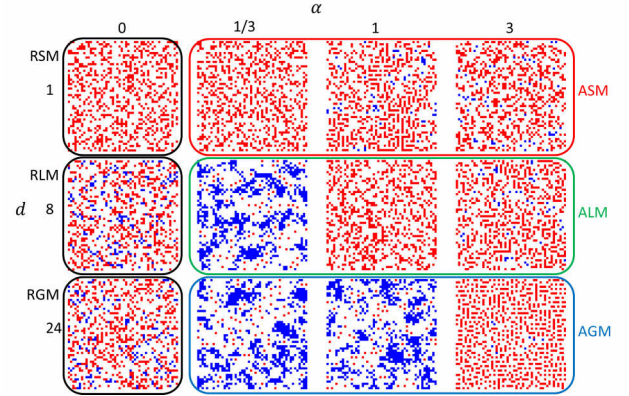


Figure 2: Typical snapshots of simulations after 10,000 generations. Blue and red dots represent cooperators and defectors, respectively. Cooperators dominated in the population in ALM ( $\alpha = 1/3$ ) and AGM ( $\alpha = 1/3, 1$ ). In those cases, cooperators successfully maintained a certain distance with defectors.  $T = 3.5$  and  $\rho = 0.3$ .

An extended version of this manuscript will soon be posted to arXiv.org.

# Testing Biological Models for Non-linear Sensitivity with a Programmability Test

Hector Zenil, Gordon Ball and Jesper Tegnér

Unit of Computational Medicine

Karolinska Institute

Centre for Molecular Medicine, Solna, Stockholm, Sweden.

{hector.zenil,gordon.ball,jesper.tegner}@ki.se

## 1 Introduction

We address the question of what kind of test could be implemented to establish whether an artificial system is living or not with a computability and programmability test as advanced in [2]. We claim that necessary conditions for life are non-linear behavioural variability and sensitivity to external stimuli.

We advance an algorithmic information concept of programmability as a combination of behavioural change and external control similar to the one presented in [1] in the context of self-assembly and non-DNA spatial computation. We look into evaluating, classifying and discriminating biological models from the BioModels Database (<http://www.ebi.ac.uk/biomodels-main/>), a centralized database of curated quantitative models of biochemical interest whose dynamical space, time-evolution and model's reaction to their “environment” (sensitivity) is studied, leading to questions such as the robustness and param-



eter orthogonality.

Similarity metrics and information distances are applied and results are presented in the direction of characterisations of the behaviour of physical and natural systems, allowing the classification of qualitative properties and the assessment of simulation results in terms of algorithmic information content with special focus on transitions, clustering, variability and space parameter discovery and behavioural mapping.

The testing approach based on a system capability to react to external stimuli and transfer information as reflected in the proposed measures of programmability indicate the susceptibility of a system to be (efficiently) programmed where we can ask questions such as whether programmability differences originate from either (a) a structural difference in the modelling equations, or (b) differences in how much parameters are allowed to vary, and thereby having a greater/smaller flexibility to respond to external stimuli. Model understanding may be hindered by the assumed dataset and the estimated parameters on which models are often heavily dependent. The approach promises to provide some further insight on different types of biological models.

## References

- [1] G. Terrazas, H. Zenil and N. Krasnogor, Exploring Programmable Self-Assembly in Non-DNA based Molecular Computing, submitted to *Natural Computing* (in 2nd. revision).
- [2] What is Nature-like Computation? A Behavioural Approach and a Notion of Programmability, *Philosophy & Technology*, 2013. <http://link.springer.com/article/10.1007/s13347-012-0095-2>

# SimSoup: Molecules Designed for Switchable Autocatalytic Memory

Chris Gordon-Smith<sup>1</sup>

<sup>1</sup>SimSoup  
c.gordonsmith@gmail.com

## Abstract

A memory system based on an artificial chemistry is presented. This is relevant for metabolism based Origin of Life theories, and in the field of biological and chemical computing. Each memory unit can be switched between three alternative active states. A unit maintains itself in a particular state using an autocatalytic reaction process. Switching between states occurs when an external stimulus triggers the autocatalytic process for the new state, along with an associated process that inhibits autocatalytic activity for the old state. Artificial molecular species with structures designed to support the autocatalytic and inhibiting processes are presented. The SimSoup artificial chemistry simulator is used to show that the structures do indeed produce the memory system behaviour. With the advent of engineering at the molecular level, it may be possible to transfer the concepts from an *in silico* environment to a chemical environment.

## Introduction

### Background And Motivation

**Inheritance at the Origin of Life.** Contemporary organisms and viruses use DNA or RNA template molecules to provide the memory needed for inheritance and evolution. Origin of Life research suggests that template molecules and the associated enzymes needed for their accurate replication are too complex to be plausible in a prebiotic environment. Some Origin of Life theories envisage *metabolism based* inheritance in which protocells or lipid enclosed droplets without template molecules reproduced by growth and division. Variations in the metabolisms of different individuals would have led to differences in fitness that would drive evolution. For this to be workable, successful variations in metabolism would have to be ‘remembered’ and passed on to offspring.

The SimSoup project is investigating memory in chemical networks that could have supported metabolism based inheritance in early organisms.

**Biological and Chemical Computing.** A key challenge for the newer field of biological and chemical computing is the development of memory systems using components that can be readily constructed and manipulated. Such systems may be used for various purposes, including the provision of

an inheritance mechanism for the evolution of artificial systems. The simple memory mechanisms being investigated by SimSoup are also relevant in this field.

## Conceptual Background

The SimSoup project takes inspiration from:

- Metabolism based Origin of Life theories including those of Aleksandr Oparin (Oparin, 1957), Stuart Kauffman (Kauffman, 1993), Freeman Dyson (Dyson, 1999), and the Lipid World theory and GARD model of Doron Lancet’s group (Segré et al., 1998, 2001a,b)
- Günter Wächtershäuser’s chemo-autotrophic Iron-Sulphur World (Wächtershäuser, 2006)
- Walter Fontana’s Algorithmic Chemistry (Fontana, 1992)
- Graham Cairns-Smith’s clay crystal and genetic takeover theory (Cairns-Smith, 1982)
- Tibor Gánti’s principles of life and chemoton theory (Gánti, 2003)
- Network theory, particularly the work of Sanjay Jain and Sandeep Krishna (Krishna, 2003)
- The Chemical Organisation Theory of Peter Dittrich and Pietro Speroni di Fenizio (Dittrich and di Fenizio, 2007)
- Linus Pauling’s chemical bond theory (Pauling, 1960).

## The SimSoup Artificial Chemistry Model

The main features of the model are outlined below. A detailed description is available elsewhere (Gordon-Smith, 2013). The program code is also available (SimSoup, 2013).

SimSoup *Molecules* are two dimensional rigid structures built from *Atoms* bonded together such that they occupy fixed positions on a square ‘Board’ (similar to a chess board). Each square contains at most one Atom. Bond angles are always either 90° or 180°, and bond lengths are all equal. Atoms bond together in a way broadly consistent with valence bond theory. Each distinct molecular structure (or species) is a *Molecule Type*.

Molecules can join or split to form Molecules of different types. Joining must respect the ‘one Atom per square’ rule, and can take place with the Molecules in any of their eight possible relative orientations. Splitting occurs by breaking the weakest set of bonds that hold the Molecule into a single unit. When two Molecules join (eg  $A + B \rightarrow C$ ), a *Construction Interaction* takes place. A splitting interaction (eg  $D \rightarrow E + F$ ) is called a *Fission Interaction*.

Bond strengths are usually fixed according to the types of Atom at each end of the bond. However, some bonds are *Perturbable*; they can be weakened or strengthened by nearby Atoms that do not themselves participate in the bond.

Construction and Fission Interactions take place in a well-mixed *Reactor*. This is the physical environment in which the Molecules exist. Interaction rates are determined by rate constants that are calculated using parameters analogous to the Arrhenius parameters used by chemists. Fissions typically proceed more quickly for Molecule Types with weak bonds. The rates of Constructions are proportional to the product of the concentrations of the reactants.

### Previous Work: A Non-Switchable Memory Bank

A previous work (Gordon-Smith, 2011), described an artificial chemistry providing a ‘memory bank’. The motivation was to provide a proof of concept demonstrating that a chemical network could have a large number of alternative states, as would be needed for metabolism based inheritance.

Each memory unit in the previous work is an *Autocatalytic Sub-Network* within the overall network, and has two states: active and inactive. Interactions occurring in the active state produce Molecules needed for further activity. Each unit corresponds to a particular monomer Molecule Type. Units are arranged in series in the network, and the shapes of their Molecule Types depend on series and position in series.

The presence or absence of monomers of particular types in the Reactor corresponds to stored information. Memory capacity depends on the number of monomer series, and on the number of monomer types in each series.

This previous memory bank has the drawback that the memory units are not easily switchable; once a unit has been activated it cannot be readily de-activated. It can be argued that such switchability may not be essential in some Origin of Life contexts<sup>1</sup>. However, it would be required in many chemical computing scenarios.

### A Switchable Memory Unit

This section presents an *in silico* chemical memory unit that can be readily switched between states. Multiple memory units can co-exist within the same physical environment (Reactor) without interfering with one another; this is due to specific features of the molecular structures.

<sup>1</sup>Although it may affect capacity for co-evolution.

### Concept For Switchability: Rock-Paper-Scissors

The design for switchable memory units is based on the game ‘Rock-Paper-Scissors’. The ability of a unit to ‘remember’ is due to the operation of its three Autocatalytic Sub-Networks. In order to make the memory unit switchable, a mechanism is introduced that enables activity in one Autocatalytic Sub-Network to cause activity in another to be inhibited.

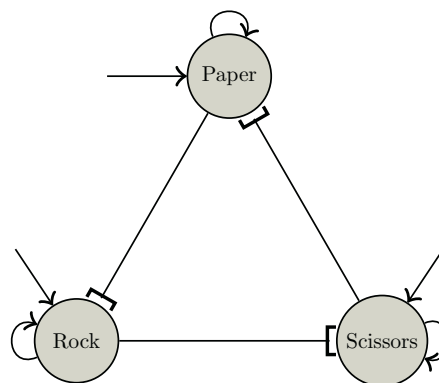


Figure 1: Rock-Paper-Scissors concept for switchable memory units

The overall concept is illustrated in Figure 1. The diagram as a whole represents a switchable memory unit with three active states.<sup>2</sup> Each circle corresponds to an Autocatalytic Sub-Network maintaining one of the states. The ‘loopback’ arrows denote the (autocatalytic) positive feedback mechanisms by which they maintain themselves. The straight arrows terminating at each circle denote external stimuli that can activate the Autocatalytic Sub-Networks. The lines between circles terminated with ‘I’ symbols signify that Molecules produced by autocatalytic activity maintaining one state can inhibit activity for another state. For example, the ‘Scissors’ Autocatalytic Sub-Network inhibits the Paper Autocatalytic Sub-Network.

Except during state transitions, only one state can remain active during any period; if two are active simultaneously then one inhibits and ‘kills’ the other. If all three were active, then all three would also be inhibited, leading to an unstable state with no clear overall ‘winner’.

In order for the memory unit to be switched, an external stimulus must be provided to activate the new state. For example, if Rock is currently active, then switching occurs when an external stimulus is provided to activate the Paper state. The system then goes through a transition in which the Rock Autocatalytic Sub-Network is inhibited by a process using Molecules produced by the growing autocatalytic activity for the Paper state. The Paper state itself is *subject*

<sup>2</sup>In the inactive or ‘dead’ state, none of the three Autocatalytic Sub-Networks would be active.

to no such inhibiting constraint. If the unit is to be switched to Scissors, then two separate switching actions must take place, first to Paper and then to Scissors.<sup>3</sup>

### Overview Of Memory Unit Design

Figure 2 illustrates the design for a memory unit. It shows both the structure of the (artificial) chemical network, and a ‘thumbnail’ structure diagram for each Molecule Type involved in maintaining and inhibiting the unit’s ‘Rock’ state. The arcs between lines joining Molecule Types identify Constructions and Fissions. Specific details of the molecular structures and bonds, and the way they support the operation of the memory unit are given later, along with naming conventions for Molecule Types.

Each co-existing memory unit has its own set of characteristic ‘Rock-Paper-Scissors’ Molecule Types and states. For example, memory unit  $n$  has possible active states,  $\text{Rock}_n$ ,  $\text{Paper}_n$ , and  $\text{Scissors}_n$ , and uses Molecule Types such as  $R_n$  and  $R_{n\text{pre}}$ . Each memory unit state has an Autocatalytic Sub-Network, and an *Inhibiting Sub-Network* for the inhibition mechanism.

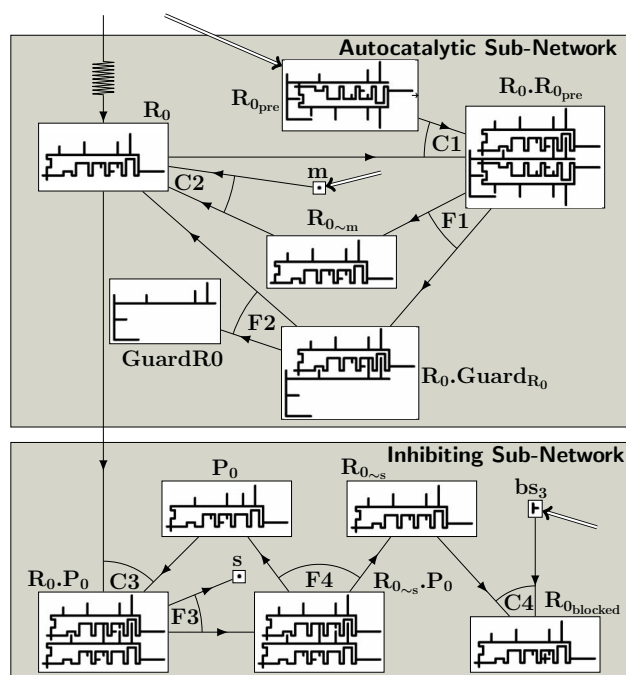
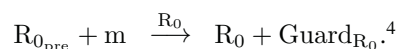


Figure 2: Overview of Network and Molecular Structures for the ‘Rock<sub>0</sub>’ State. The top section shows Molecule Types and Interaction Types for the Rock<sub>0</sub> Autocatalytic Sub-Network. The lower section shows the Inhibiting Sub-Network for Rock<sub>0</sub>. The zig-zag arrow indicates an external stimulus. Double arrows indicate a constant supply. Labelled arcs indicate Constructions and Fissions.

<sup>3</sup>The need for two stage switching is a drawback, but a consequence of the simple design in which the same input is used both for activating one sub-network and inhibiting another.

**The Autocatalytic Sub-Network.** The zig-zag arrow terminating at  $R_0$  indicates that an external stimulus consisting of a few  $R_0$  Molecules can be provided to the memory unit. The double arrows terminating at  $R_{0\text{pre}}$  and  $m$  (Metal) indicate that a constant supply of each is provided.

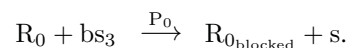
The autocatalytic reaction process begins when  $R_{0\text{pre}}$  and  $R_0$  combine under Construction C1 to form  $R_0.R_{0\text{pre}}$ . Each  $R_0.R_{0\text{pre}}$  Molecule then breaks up in two successive Fissions, F1 and F2. F1 produces one Molecule of  $R_{0\sim m}$  and one of  $R_0.\text{Guard}_{R_0}$ . The  $R_{0\sim m}$  Molecule goes on to combine with an ‘m’ Atom (Metal) in Construction C2 to return an  $R_0$  Molecule. F2 splits the  $R_0.\text{Guard}_{R_0}$  Molecule into one Molecule of  $\text{Guard}_{R_0}$ , and a second (excess)  $R_0$  Molecule. The overall reaction process is:



The process can only start if an initial input of  $R_0$  is provided as a catalyst. Once started, the process produces an excess of  $R_0$ , and no further external input of  $R_0$  is needed.

**The Inhibiting Sub-Network.** The Inhibiting Sub-Network impedes activity of the Autocatalytic Sub-Network by converting  $R_0$  Molecules to  $R_{0\text{blocked}}$  Molecules that cannot participate in the autocatalytic reaction process. The conversion starts with Construction C3, which produces  $R_0.P_0$  from  $R_0$  and  $P_0$ . Each  $R_0.P_0$  Molecule then splits (Fission F3) to produce an Atom of  $s$  (Stoppite) and an  $R_{0\sim s}.P_0$  Molecule. Each  $R_{0\sim s}.P_0$  Molecule splits (Fission F4) to produce a Molecule of  $R_{0\sim s}$  and a Molecule of  $P_0$ , returning the one required for Construction C3. Finally, the  $R_{0\text{blocked}}$  Molecule is produced by combining the  $R_{0\sim s}$  Molecule with a  $bs_3$  Molecule (Construction C4).

The overall reaction process for the Inhibiting Sub-Network is:



The process can only proceed in the presence of  $P_0$  as a catalyst. The Autocatalytic Sub-Network for the Paper<sub>0</sub> state produces  $P_0$ , and this is the basis by which the Paper<sub>0</sub> inhibits the Rock<sub>0</sub> state.

### Molecular Structures and Bonds

This section details the molecular structures used by the memory unit, and the way they support its operation. The structures are constructed from (artificial) Atoms that bond in particular ways, and so the Atom Types and Bond Types are described first.

#### Atom Types

The memory unit uses Atom Types with properties designed to support the Interaction Types that are needed for its operation. Table 1 shows these Atom Types. The ‘Bonds’ col-

<sup>4</sup> $\text{Guard}_{R_0}$  produces further ‘waste’ products (not shown).



um indicates the number of bonds that each Atom Type can form. All the Atom Types have mass 10.

| Name/Symbol    | Bonds | Usage / Capability                                                        |
|----------------|-------|---------------------------------------------------------------------------|
| Assemblite (●) | 2     | Used to build framework of molecular structures                           |
| Blockite (b)   | 4     | Used to build bs <sub>3</sub> 'Blocker'                                   |
| Hookite (h)    | 4     | Used to provide a 'hook' bonding site                                     |
| Junctium (j)   | 3     | Used to provide a 3 way junction in a structure                           |
| Loosium-1 (w)  | 2     | Provides a weak (loose) bonding site for Loosium-2                        |
| Loosium-2 (x)  | 2     | Provides a weak (loose) bonding site for Loosium-1                        |
| Metal (m)      | 1     | Can perturb nearby Perturbium bonds, even though not bonded to Perturbium |
| Perturbium (p) | 3     | Produces bonds that can be weakened or strengthened by nearby Metal atoms |
| Stoppite (s)   | 1     | Stops further growth of the Molecule at a site                            |

Table 1: Atom Types used for the switchable memory units

## Bond Types

Table 2 shows the Bond Types that can exist between the various Atom Types. The meanings of the cell entries are as follows:

- Blank: Atom of these types do not bond
- x: Atoms bond with Enthalpy (strength) 100
- w: Atoms bond weakly, with Enthalpy 10
- p: Perturbable Bond

|   | a | h | j | b | m | p | s | w | x |
|---|---|---|---|---|---|---|---|---|---|
| a | x |   |   |   |   |   |   |   |   |
| h | x |   |   |   |   |   |   |   |   |
| j | x | x | x |   |   |   |   |   |   |
| b |   |   |   |   |   |   |   |   |   |
| m | x | x | x |   |   |   |   |   |   |
| p | x | p | x | x |   |   |   |   |   |
| s | x | x | x | x |   | p | x |   |   |
| w | x |   | x |   |   |   |   |   |   |
| x | x |   | x |   |   |   |   | w |   |

Table 2: Bond Types: Each cell shows the characteristics of a bond between a top row Atom and a left column Atom.

A *Perturbable Bond* is one whose strength can be affected by the presence of a nearby Atom of a specific type, even though that Atom is not bonded to either of the Atoms participating in the bond. For the work described in this paper, the normal Enthalpy of the Perturbable Bonds shown in the table is 100, but reduces to 5 in the presence of a nearby Metal Atom, making the bond more likely to break quickly.

## Molecule Types

This section shows details of the Molecule Types for the memory unit, and describes the way they support its operation. Conventions for naming Molecule Types are described where they are first used.

**Core Molecule Types: R<sub>n</sub>, P<sub>n</sub>, S<sub>n</sub>.** Figure 3 shows the *Core Molecule Types* for the three Autocatalytic Sub-Networks of the n=0 memory unit. R<sub>0</sub> is produced by the Rock<sub>0</sub> Autocatalytic Sub-Network. Similarly P<sub>0</sub> and S<sub>0</sub> are produced by the Paper<sub>0</sub> and Scissors<sub>0</sub> Autocatalytic Sub-Networks respectively.

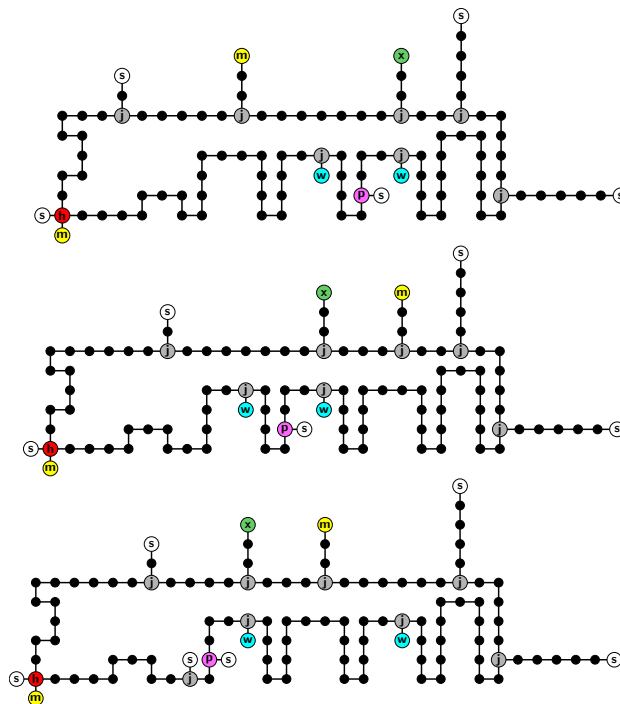


Figure 3: R<sub>0</sub> (top), P<sub>0</sub> (centre) and S<sub>0</sub> (lower): The Core Molecule Types for the Rock<sub>0</sub>, Paper<sub>0</sub> and Scissors<sub>0</sub> Autocatalytic Sub-Networks.

The following features can be noted:

- Each Molecule Type has various recesses and protuberances. These are positioned to prevent Constructions that would produce Molecule Types that interfere with the operation of the memory unit
- S<sub>0</sub> is one unit wider than P<sub>0</sub>, which is one unit wider than R<sub>0</sub>. This is also to prevent the production of interfering Molecule Types
- The three central recesses along the bottom of each of R<sub>0</sub>, P<sub>0</sub> and S<sub>0</sub> are key to the mechanisms for both autocatalysis and inhibition. The recess that includes both a Loosium-1 (w) Atom and a Stoppite (s) Atom is called the

**Owner Recess.** The recess that includes just a Loosium-1 (w) Atom is called the *Successor Recess*, and the ‘empty’ recess is called the *Predecessor Recess*

- The recesses are arranged cyclically, so that the Successor Recess for one Molecule Type lines up with the Owner Recess for the Molecule Type that ‘beats’ (or succeeds) it
- Above each Owner Recess there is a protuberance (the *Owner Protuberance*) that terminates with a Loosium-2 Atom. Above each Predecessor Recess there is a *Predecessor Protuberance* that terminates with a Metal Atom.

The roles of these protuberances and recesses in the operation of the memory unit are described later in this section.

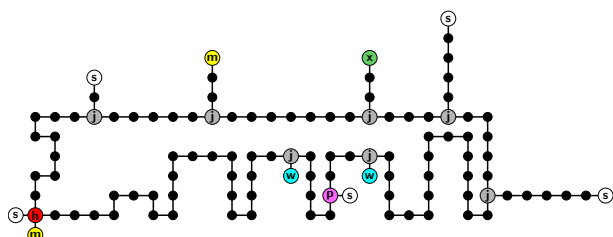


Figure 4:  $R_1$ , the Core Molecule Type for  $Rock_1$ .

Figure 4 shows  $R_1$ , the Core Molecule Type for the  $Rock_1$  Autocatalytic Sub-Network. It is identical to  $R_0$ , except that there is an additional space to the right of the Owner Recess.  $P_1$  and  $S_1$  have similar additional space by comparison with their  $n=0$  counterparts. Core Molecule Types for larger  $n$  values have correspondingly larger spaces. The spacing ensures that corresponding Molecules (eg  $R_0$ ,  $R_1$ ) for different co-existing memory units do not interfere.

**‘Missing Atom’ Variants of Core Molecules.** Variants of the Core Molecule Types with a ‘missing’ Metal or Stop-pite Atoms occur in the Autocatalytic and Inhibiting Sub-Networks. If the Stop-pite Atom in the Owner Recess of the  $R_0$  Molecule shown at the top of Figure 3 is removed, the result is called  $R_{0\sim s}$ . Similarly, if the Metal Atom at the bottom left of the Molecule is removed, the result is  $R_{0\sim m}$ .

**Core Molecule Type Precursors and Guards.** The Autocatalytic Sub-Network for a memory unit is provided with a constant supply of *Precursors* for its Core Molecule Types.

Figure 5 shows  $R_{0\text{pre}}$ , the Precursor for  $R_0$ . This consists of  $R_{0\sim m}$  ( $R_0$  less a Metal Atom) with an adjoined *Guard*. The Guard covers the Owner, Successor and Predecessor Recesses for the Core Molecule Type, making them inaccessible to protuberances. Figure 6 shows  $\text{Guard}_{R_0}$ , the Guard for  $R_0$ .

**Role Of The Precursor.** A type  $n$  memory unit is provided with a supply of  $R_{n\text{pre}}$ ,  $P_{n\text{pre}}$  and  $S_{n\text{pre}}$ , the Precursors

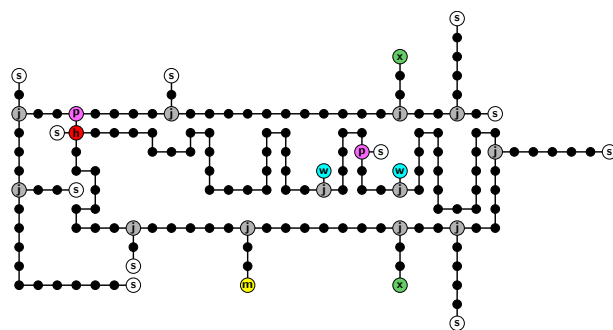


Figure 5:  $R_{0\text{pre}}$ , the Precursor for  $R_0$ .

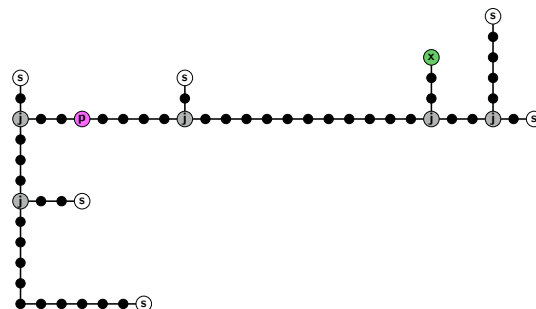


Figure 6:  $\text{Guard}_{R_0}$ , the Guard for  $R_0$ .

for its Core Molecule Types  $R_0$ ,  $P_0$  and  $S_0$ . The Precursor can be split as described shortly by the Autocatalytic Sub-Network to release a Core Molecule. The steady supply of Precursor Molecules therefore provides the ‘food’ that enables a memory unit to maintain its active state.

**Role Of The Guard.** The Guard part of a Precursor Molecule Type has two roles:

- It has a Protuberance terminated with a Loosium-2 (x) Atom that bonds weakly to the Loosium-1 (w) Atom in the corresponding Core Molecule Type to form a Precursor Splitter Complex
- It prevents Precursors from combining with successor Core Molecules to produce Precursor.Core Complexes instead of Core.Core Complexes that are essential for the Inhibiting Sub-Network.

**The Precursor Splitter Complex.** A Core Molecule Type can join with its Precursor to produce a *Precursor Splitter Complex*. Figure 7 shows the Precursor Splitter Complex,  $R_0.R_{0\text{pre}}$ , for the  $Rock_0$  Autocatalytic Sub-Network.

The two Molecules join at the Loosium-1 to Loosium-2 (w-x) bond in the Core Molecule’s Owner Recess (see Construction C1 in Figure 2). The Metal Atom at the bottom left of the Core Molecule is placed just above (but not bonded to) the Perturbium-Hookite bond at the top left of the Precursor.

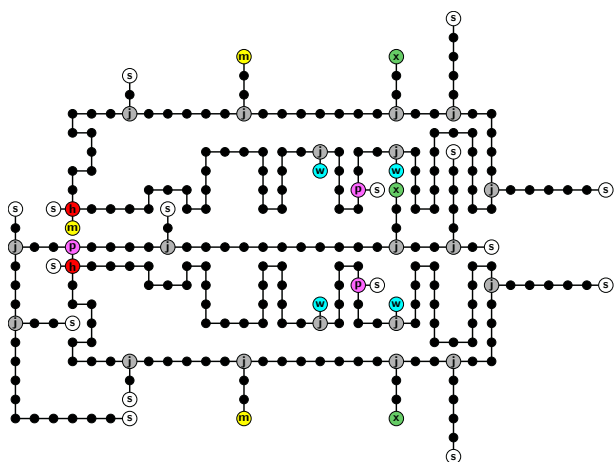


Figure 7:  $R_0.R_{0_{pre}}$ : The Precursor Splitter Complex for the  $Rock_0$  Autocatalytic Sub-Network

This weakens the bond, which soon splits (Fission F1) releasing a ‘missing Metal’ variant of the Core Molecule and a Reverse Guard Complex (see next section).

The ‘missing Metal’ Core variant combines with freely available Metal to form a Core Molecule. This replaces the Core Molecule that combined with the Precursor.

**The Reverse Guard Complex.** When a Precursor Splitter Complex splits, one of the products is a *Reverse Guard Complex*. This is a Core Molecule with a Guard in a ‘reverse’ configuration. Figure 8 shows the Reverse Guard Complex,  $R_0.Guard_{R_0}$ , for the  $Rock_0$  Autocatalytic Sub-Network.

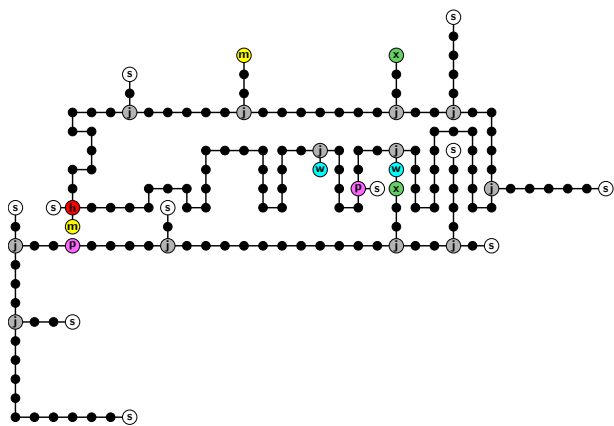


Figure 8:  $R_0.Guard_{R_0}$ , the Reverse Guard Complex for the  $Rock_0$  Autocatalytic Sub-Network

The Reverse Guard Complex is held together by the Loosium-1 to Loosium-2 (w-x) bond that was formed when the Core Molecule joined with a Precursor in Construction C1. This is a weak bond (see Table 2), and soon breaks (Fission F2), releasing a Core Molecule and an (un-needed)

Guard. The Core Molecule released here is the one that previously combined with the Precursor. A Core Molecule was also produced by Fission F1, and so there is an excess.

**The Core.Core Complex.** Two Core Molecule Types for different states of the same memory unit can join to form a *Core.Core Complex* that plays a key role in the Inhibiting Sub-Network. Figure 9 shows the  $R_0.P_0$  Core.Core Complex.

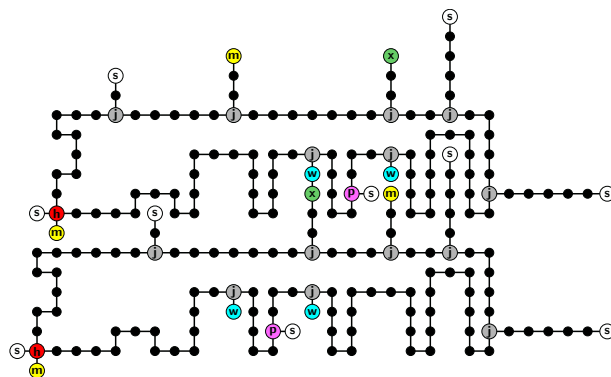


Figure 9:  $R_0.P_0$ : The Core.Core Complex for the  $Rock_0$  Inhibiting Sub-Network

The two Core Molecules are joined at the weak Loosium-1 to Loosium-2 (w-x) bond in the top Molecule’s Successor Recess, and will soon break apart again (Fission F4). However, while the complex persists, a Metal Atom from the lower (Successor) Molecule is held next to the Perturbium to Stoppite bond in the top Molecule’s Owner Recess. This bond is weakened, and the Stoppite Atom soon splits off (Fission F3). This makes available a bonding site that will be used to block autocatalysis.

**Blocked Core Molecule Types and bs3 Blockers.** The Inhibiting Sub-Network converts Core Molecules to *Blocked Core Molecules* that cannot participate in the autocatalytic reaction process.

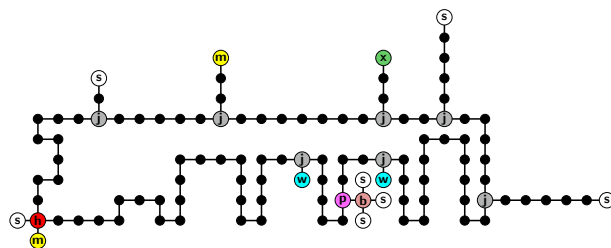


Figure 10:  $R_{0_{blocked}}$ : the Blocked Core Molecule for the  $Rock_0$  Inhibiting Sub-Network

The blocking is achieved by incorporating a small *Blocker* Molecule,  $bs_3$ , into the Core Molecule’s Owner Recess.

This is done by removing a Stoppate Atom from a Core Molecule to produce a 'missing Stoppate' variant (Fissions F3 and F4), and attaching a freely available Blocker at the bonding site that this creates (Construction C4).

Figure 10 shows  $R_{0\text{blocked}}$ , the Blocked Core Molecule Type for  $R_0$ . The small  $bs_3$  Blocker can be seen in the Owner Recess.

## Results

Figure 11 shows the operation of a 9-state memory system comprising two 3-state memory units co-existing in the same Reactor. The scenario was as follows:

- Atom Types and Bond Types as described earlier
- A constant supply of Precursors for the  $n = 0$  memory unit ( $R_{0\text{pre}}$ ,  $P_{0\text{pre}}$  and  $S_{0\text{pre}}$ ) is provided, 400 of each every 10 timesteps
- A similar supply of  $n = 1$  Precursors is setup
- A constant supply of Metal Atoms and  $bs_3$  Blockers is setup, 2000 of each every 10 timesteps
- A small stimulus of 7  $R_0$  Molecules is input at time 10000, and at subsequent intervals of 60,000 timesteps. Similar series of stimuli are setup for  $P_0$  and  $S_0$ , starting at times 30,000 and 50,000 respectively
- Stimuli are setup for the  $n = 1$  Core Molecules at times 10,000, 70,000 and 130,000
- Each Molecule has a removal probability at each timestep of 0.005.

The top plot shows that following the  $R_0$  stimulus at time 10,000, the number of  $R_0$  Molecules in the Reactor rises from zero to about 850, and stays at that level. This is due to the operation of the  $Rock_0$  Autocatalytic Sub-Network. At time 30,000 the  $P_0$  stimulus is added. The number of  $P_0$  Molecules rises to about 850 due to the operation of the  $Paper_0$  Autocatalytic Sub-Network, and the number of  $R_0$  Molecules falls to zero due to the operation of the  $Rock_0$  Inhibiting Sub-Network. The rest of the plot shows repeated cyclic switching between states of the  $n = 0$  unit.<sup>5</sup>

The middle plot shows the numbers of Core.Core Complex Molecules. These are key to the operation of the Inhibiting Sub-Network during the transition from one state of the memory unit to another. As the plot shows, they are present only during these transitional periods.

The lower plot shows the numbers of Core Molecules for the co-existing  $n = 1$  memory unit. The top and lower plots together show that the system as a whole has 9 states, corresponding to the number of possible states for two independently switchable 3-state units.

<sup>5</sup>The small excess of  $P_0$  at time 100,000 is as yet unexplained.

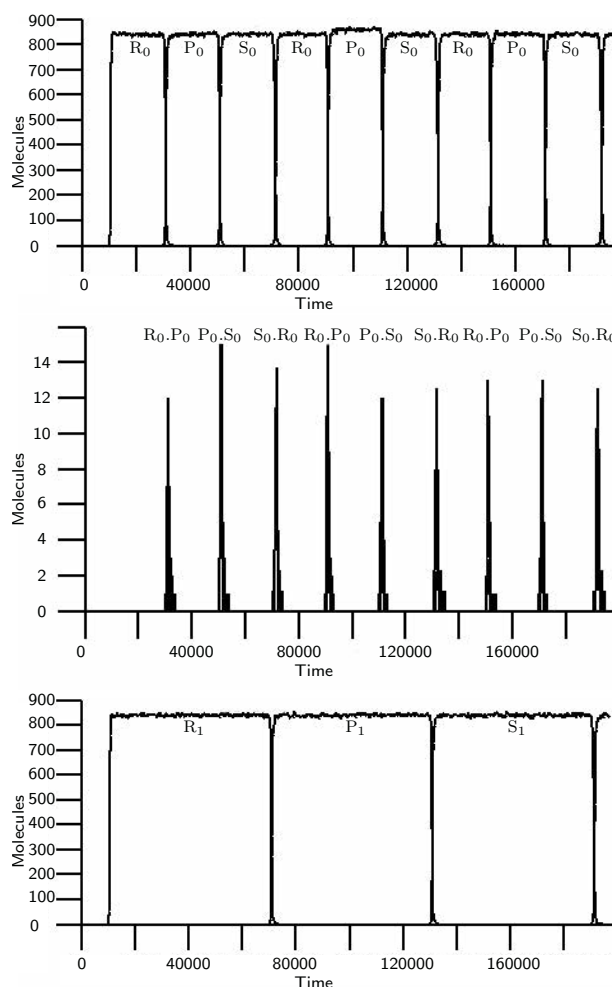


Figure 11: Operation of two co-existing 3 state memory units. Top: The  $n = 0$  unit cycling through its three states. Middle: Numbers of Core.Core Complexes present during switching of the  $n = 0$  unit. Bottom: The  $n = 1$  unit cycling through its three states.

## Conclusions

1. It has been demonstrated that it is possible to produce a switchable memory unit using (artificial) Molecules with structures specifically designed to produce a reaction network with Autocatalytic and Inhibiting Sub-Networks
2. It is possible to combine multiple memory units in the same (well mixed) chemical environment in such a way that they do not interfere with each other's operation. In the results presented here, two 3-state units were combined to produce a memory system with 9 states
3. The design of the Molecules is such that further memory units with higher  $n$  values can be produced by new Core Molecule Types with additional spaces to the right of the Owner Recess. The number of states for a system including  $m$  memory units is  $3^m$ , and the number



of Core Molecule Types needed is  $3m$ . A memory system with 30 Core Molecule Types could therefore support  $3^{10} = 59049$  states. The number of states for a system with 60 Core Molecule Types would be almost 3.5 million

4. From an Origin Of Life perspective, a population of protocells can be envisaged, each containing a memory system in a particular state. The memory system would influence other aspects of a larger reaction network affecting behaviour, such that it could support an evolutionary process in which fitter protocells were selected.

It can be argued that an inheritance mechanism supporting 3.5 million alternative ‘phenotypes’ provides sufficient scope for such an evolutionary process to get a foothold. This would require a protocell to have a foodset including 60 Core Precursor Molecule Types. This requirement may not be as stringent as appears at first sight; the Core Molecule Types and the Precursors all follow the same basic pattern. It can be envisaged that they could be produced by a systematic process with scope for random variations within the framework of that pattern

5. From the perspective of chemical and biological computing systems, it may be possible, with the advent of engineering at the molecular level, to use the concepts described here as the basis of a similar artificial evolutionary mechanism, or as memory for different applications.
6. The molecular structures shown in this paper are just one way of implementing the memory units. It is likely that there are many alternatives with totally different structures. Molecular structure design could take account of considerations regarding what can be easily engineered
7. There is a trade-off between the chemical complexity involved in a larger number of Core Molecule Types required for co-existing memory units vs physically separate memory units but fewer Core Molecule Types. For droplet (or ‘protocell’) based artificial evolution, the former is likely to be preferred, as it does not require physical separation of different memory units within the droplet.

## Prospects

The author would like to hear from anyone interested in transferring the concepts described here from an *in silico* environment to a chemical environment.

## References

- Cairns-Smith, G. (1982). *Genetic takeover*. Cambridge University Press.
- Dittrich, P. and di Fenizio, P. S. (2007). Chemical organisation theory. *Bull. Math. Biol.*, 69(4):1199–1231.
- Dyson, F. (1999). *Origins Of Life*. Cambridge University Press.
- Fontana, W. (1992). Algorithmic chemistry. In *Artificial Life II*. Addison Wesley.
- Gánti, T. (2003). *Principles of Life*. Oxford University Press.
- Gordon-Smith, C. (2007). Evolution without smart molecules. ECAL workshop paper. Available at <http://www.simsoup.info/Publications.html>.
- Gordon-Smith, C. (2009a). The origin of life: A network oriented view. In *Proceedings - Levels of Selection and Individuality in Evolution: Conceptual Issues and the Role of Artificial Life Models, September 2009*.
- Gordon-Smith, C. (2009b). SimSoup: Artificial chemistry meets Pauling. In *Advances in Artificial Life: 10th European Conference, ECAL 2009, Proceedings, Lecture Notes in Computer Science*. Springer-Verlag.
- Gordon-Smith, C. (2011). Non-template molecules designed for open-ended evolution. In *Advances in Artificial Life, ECAL 2011 Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*.
- Gordon-Smith, C. (2013). The simsoup guide, version 1.0. <http://www.simsoup.info/Publications.html>.
- Kauffman, S. A. (1993). *The Origins Of Order*. Oxford University Press.
- Krishna, S. (2003). *Formation And Destruction of Autocatalytic Sets in an Evolving Network Model*. PhD thesis, Indian Institute Of Science.
- Oparin, A. (1957). *The Origin Of Life On The Earth*. Oliver And Boyd.
- Pauling, L. (1960). *The Nature Of The Chemical Bond*. Cornell University Press.
- Segré, D., Ben-Eli, D., Deamer, D. W., and Lancet, D. (2001a). The lipid world. *Origins Life Evol. B.*, 31:119–145.
- Segré, D., Lancet, D., Kedem, O., and Pilpel, Y. (1998). Graded autocatalysis replication domain (GARD). *Origins Life Evol. B.*, 28:501–514.
- Segré, D., Shenhav, B., Kafri, R., and Lancet, D. (2001b). The molecular roots of compositional inheritance. *Journal Of Theoretical Biology*, 213:481–491.
- SimSoup (2013). Simsoup simulator source code. <http://code.google.com/p/simsoup/>.
- Wächtershäuser, G. (2006). From volcanic origins of chemoautotrophic life to bacteria, archaea and eukarya. *Philos Trans R Soc London [Biol]*, 361:1787–1808.

# The emergence of life and evolution: Towards a categorical approach

Roland Cazalis

SPS/URBV, University of Namur, Belgium  
roland.cazalis@fundp.ac.be

## Abstract

Among the list of proposed questions in “Open Problems in Artificial Life” (Bedau et al. 2000) the first, “how does life arise from the nonliving?” is certainly one of the most difficult to answer. The first and main challenge in the understanding of the way that life emerged from the nonliving comes from our lack of knowledge of the living organism and paradoxically, the limitations which we impose to it, so that what we call organism remains a stillborn entity at best. The second challenge is creating a formal language suited to represent the organism while taking into account all its quantitative and qualitative characteristics.

Consequently, to progress in the understanding of the organism, it is first essential to integrate two data. The first one being that both the questions “what is the living?” and “how does the living arise?” are connected. In other words, before being able to apprehend correctly the emergence of life, it would be necessary to know how to model it, almost through a theoretical biology approach. The second datum is related to finding an appropriate language which is capable of describing the organism in all its dimensions.

In this present work, we want to find answers to the first list of Bedau's questions through a more integrative definition of the living which enables us to think of its mode of emergence from the nonliving in a coextensive way. For this, we propose a categorical model of the organism. This seems to be the most suitable language at the current stage of knowledge in describing such a phenomenon.

To formalise our proposal, we use a branch of mathematics called category theory. A category is a structure that appears like a directed graph, which consists of objects and arrows between them. However, in a category, two paths from A to B may be declared equivalent in a commutative diagram. Beside, a functor is a mapping from one category to another that preserves the structures.

As far as our main purpose is to unit entities in a special bigger one, this requires encoding interfaces, functional relationships, and transfer of data, extensive and intensive properties. In short, to convert an organism appearance into the ontology of category theory, we need concepts that allow us to group and lay out things. These concepts are best

captured by pushout and its dual pullback expressed in terms of diagrams of arrows, which are special types of more general constructs known as colimit and limit respectively.

In a category  $\mathcal{C}$ , the pushout (figure 1) for two morphisms  $f: A \rightarrow B$  and  $g: A \rightarrow C$  is an object  $D$ , and two morphisms  $j: B \rightarrow D$  and  $k: C \rightarrow D$ , such that the square commutes ( $f \circ g = j \circ k$ ); and for any objet  $D'$  and pair of morphisms  $j': B \rightarrow D'$  and  $k': C \rightarrow D'$  satisfying  $j' \circ f = k' \circ g$ , there is a unique morphism  $h: D \rightarrow D'$  such that diagram commutes. A dual is obtained by reversing the arrows. A complete presentation of category theory can be found in Lawvere and Schanuel (1997), and Ehresmann and Vanbremeresch (2007).

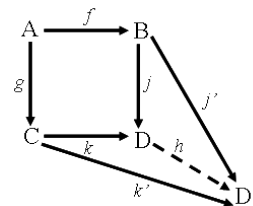


Figure 1

Pushout diagram

Three intuitions guide our approach of the living. First: a living organism transcends its material template. Second: it is a hierarchical multi-scale system within an environment (Cottam et al. 2004). Third is the idea of closure or autonomy, which was extensively analysed *inter alia* by Robert Rosen (1991). We argue that these different intuitions and characteristics of an organism are best described through the importance of colimits, functors or identity arrows. From a single cell to a complex multicellular organism such as a plant, a repetitive multi-scale hierarchical concept is found embedded, the one in the other, giving the hierarchical system a self-similarity dimension.

An organism appears with the advent of the closure of the system. From the categorical viewpoint, it happens with the instantiation of a functor called the “quale identity”. The quale can be illustrated by a diagram using the categorical idea of identity arrow. Although there are other arrows with the same object for domain and co-domain, the identity arrow defines a unique relationship of the object with itself.

The dual of the quale identity is represented by the hierarchical multi-scale system. While the colimit, which is a bottom-up process, constructs the system, the quale identity is a top-down process which consolidates it. At first, the final system appears as a resulting entity of a pullback and a pushout fitting. The difference between these two characteristics is that the hierarchical system has a quantitative nature, whereas the quale identity is a qualitative function. To summarise, let  $(a)$  be a hierarchical system in its environment  $(A)$ , and its identity quale  $(I_{qa})$ , an organism can be formalised as  $I_{qa}(a, A)$ .

The hierarchical system and the quale identity work as two semi-categories. If we reduce the hierarchical system to a collective link  $(L)$ , as the quale identity is a reduction of the top-down process, both morphisms become endofunctors. Their correlation depends on natural transformations, which are morphisms between functors. From these data, a model of the organism can be represented by the figure 2.

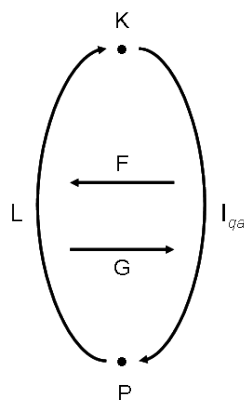


Figure 2

Formal categorial representation of an organism where  $(K)$  is the colimit of the multi-scale hierarchical system  $(a)$ .  $(P)$  is the pattern of subsystems of  $(a)$ .  $(I_{qa})$  is the identity quale of  $(a)$ , and  $(L)$  is a set of links from  $(P)$  to  $(K)$ . The dualistic aspect of the entity is overcome by a set of natural transformations  $(F, G)$  which interplay between  $(I_{qa})$  and  $(L)$  and by which the organism becomes an evolutionary unit (according to Cazalis, 2013).

With this model of the organism, we can better comprehend its mode of emergence from the nonliving. We propose that in a given environment, there are arguably two essential events in the advent of organisms. One is the formation of the multi-scale hierarchical system having the characteristics  $(C_T)$ , and the instantiation of the quale identity within it. The second event constitutes a real heart of the system, giving it autonomy and its robustness, which are the first characteristics of the individual. It is not necessary that the hierarchical system be perfectly organised; it is actually preferable that it has certain plasticity. Indeed, the quale identity exercises a canopy function within the organism. The canopy function has two main aspects. First, it is a function

which comes from the very system itself and characterises the quale. Secondly, as a functional image of the hierarchical system, it has a compensating role especially when the system has not reached its fluctuating balance; that is its coherence. On the same note, we assume that the characteristics  $(C_T)$  do not mean a full coherent system because once the quale identity is instantiated; it sustains and perfects the integration.

The interaction between the endofunctors  $(L)$  and  $(I_{qa})$  allows the organism to be an evolutionary unit. Changes in  $(L)$  from modifications to the environment or by natural destruction of its elements entails to their replacement or substitution to preserve the feature memorized in the quale.

In the same way, the quale could be subject to positive or negative interferences with other organisms present in the environment. As long as the characteristics of the hierarchical system are protected, the maintenance of the initial quale is guaranteed.

The interactions operated by a set of natural transformations entail an internal temporality, which constitutes the second characteristic of the organism. Depending on the evolution of the organism the area of the natural transformations is subjected to a process of complexification.

Finally, how does life arise from the nonliving is suggested by the very nature of the organism according to our model. Subsequently, this work analyses the quantitative and qualitative aspects of the organism; and the notions of autonomy and internal temporality. It proposes a composition rule for the quale identity in collective organisms and discusses the problem of critical threshold  $(C_T)$  with respect to the bottom-up approach that aims to assemble artificial cells from scratch using nonliving materials.

## References

- Bedau, M. A.; McCaskill, J. S.; Packard, N. H.; Rasmussen, S.; Adami, C.; Green, D. G.; Ikegami, T.; Kaneko, K.; and Ray, T. S. (2000). Open problems in artificial life. *Artificial Life* 6:363-376.
- Cazalis, R. (2013). The organism and its double: An approach to the individual. *Progress in Biophysics and Molecular Biology*. <http://dx.doi.org/10.1016/j.pbiomolbio.2013.03.006>
- Cottam, R., Ranson, W. and Vounckx, R. (2004). Autocreative Hierarchy I: Structure -Ecosystemic Dependence and Autonomy. *SEED Journal*, 4: 24-41.
- Ehresmann, A.C., Vanbreemersch, J.-P. (2007). *Memory Evolutionary Systems: Hierarchy, Emergence, Cognition*. Elsevier Science.
- Lawvere, F. W.; Schanuel, S. H. (2009). *Conceptual Mathematics: A first introduction to categories*. Cambridge University Press.
- Rosen, R. (1991). *Life Itself*. Columbia, UP, New York.



# Using novel bio-inspired principles to improve adaptability of evolutionary robots in dynamically changing environments

Yao Yao<sup>1,2</sup>, Kathleen Marchal<sup>2,3</sup> and Yves Van de Peer<sup>1,2</sup>

<sup>1</sup>Department of Plant Systems Biology, VIB, Technologiepark 927, 9052 Ghent, Belgium

<sup>2</sup>Department of Plant Biotechnology and Bioinformatics, Ghent University, Technologiepark 927, 9052 Ghent, Belgium

<sup>3</sup>Department of Microbial and Molecular Systems, CMPG, Katholieke Universiteit Leuven, Leuven, Belgium

## Abstract

An important goal in the field of evolutionary computation and robotics is the development of systems that show self-adaptation in dynamically changing environments. However, adaptation to a changing or fluctuating environment is challenging and usually requires a dynamic solution. Here, we present a bio-inspired, agent-based controller driven by an artificial gene regulatory network that enhances the adaptability of robots in dynamically changing environments.

## Artificial genome

The Genome consists of two kinds of genes. The 'regulatory' gene is in charge of the regulatory functions while the 'structural' gene determines the phenotype of the robot through its gene expression product. The interaction between genes and the environment is based on the transcription factor binding process.

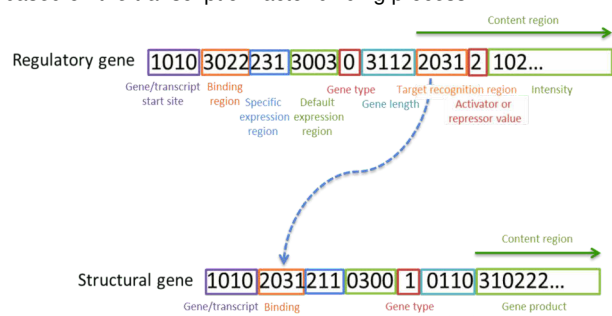
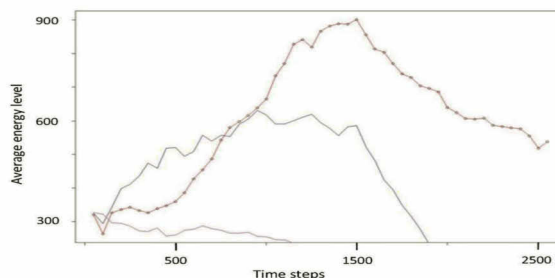


Figure 1. Artificial genome encoding the regulatory network. Any gene, irrespective of its type, consists of the following components: a transcription start site, a gene identifier (type), a gene length region, a binding site region, an expression level region (default and gene-specific expression region), and a content region which is different for structural, regulatory and signaling genes.

## Results & Conclusion

To demonstrate the advantages of our bio-inspired controller, we also developed an artificial life simulation platform that can provide a dynamically changing environment. Simulation results show that the bio-inspired GRN based controller has a better average adaptability than an evolutionary artificial neural network (ANN) based controller and a signal-based controller when placed in the same dynamically changing environments. Based on the experimental results, we believe the dynamics and the feedback loops of the GRN have great potential to enhance the adaptability of individual organisms in a changing environment.



## Agent based GRN

The dynamics of the GRN, encoded by the artificial genome, and its interaction with the environment, are modeled by an agent-based system.

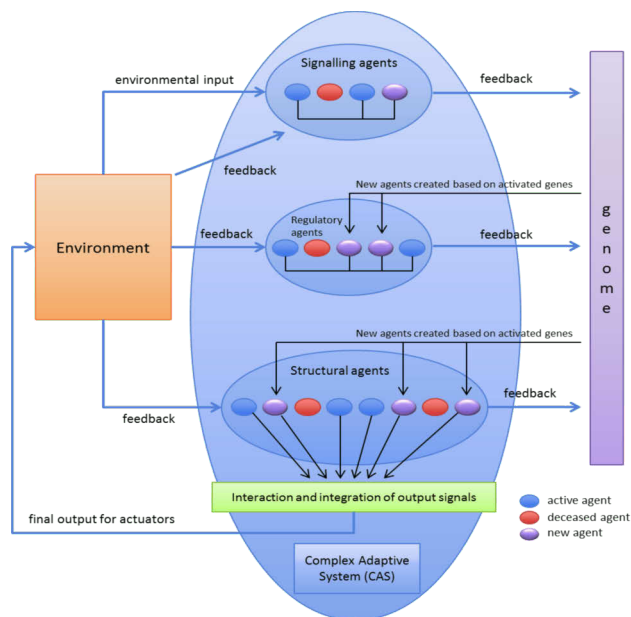


Figure 2. Agent based system modeling the condition dependent instantiation of the GRN encoded by the artificial genome. Three different kinds of agents are distinguished, namely signalling agents modeling the interaction between the environment and the artificial genome, regulatory agents that constitute the active part of the GRN encoded by the artificial genome, and structural agents that translate the encoded information of a structural gene to an output signal, which drives the actuators (e.g. wheel) of the robot. The agents based level is also essential in establishing the feedback from the environment to the system through the agents adaptability values which affect both the agents lifetime, genomic encoding, and mutation rate.

Figure 3. Average energy levels for the three different controllers during runtime. The Y-axis represents the average energy level of the robots while the X-axis represent running time measured in time steps. The red line shows the result for a single simulation experiment based on the GRN controller; the blue curve shows the result for the experiment based on the ANN controller, and the purple curve shows the result for the experiment based on the simple non-evolvable signal-based controller.



# CARDINAL-E: AIS Extensions to CARDINAL for Decentralised Self-Organisation for Network Security

Peter Scully<sup>1</sup>, Jingping Song<sup>1,2</sup>, Jules Pagna Disso<sup>3</sup> and Mark Neal<sup>1</sup>

<sup>1</sup>Department of Computer Science, Aberystwyth University, Wales SY23 3DB  
{pds7;jis17;mjn}@aber.ac.uk

<sup>2</sup>Software College, Northeastern University, Liaoning, China 110819

<sup>3</sup>Cyber Security Research Lab, EADS Innovation Works, Newport, Wales NP10 8FZ  
julesferdinand.pagna@eads.com

## Extended Abstract

This paper extends the CARDINAL architecture by Kim et al. (2005) to CARDINAL-E. CARDINAL-E keeps the innate immune system behaviour at every computer on the network and relocates the adaptive immune system behaviour to higher performance computers. Two paradigmatic shifts are achieved by this modification. First is the shift from standalone to supportive, otherwise considered as architecturally static to dynamic. This leads to an additional layer of homeostasis at a network-wide level. The intended effect is to leverage unused capacity on networks of heterogeneous machines. Secondly, the change represents a subtle granular shift from “each computer has identical immune system components” to “the network (as a whole) carries all the immune system components”. This is a synthetic network-wide “body” where organs (CARDINAL’s Periphery and Lymph Node components) are finite and proportionate in quantity, and evolve their behaviour over time.

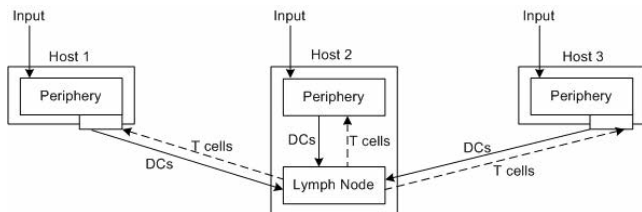


Figure 1: A simplified illustrative example of the CARDINAL-E architecture in a small network. Inputs are received at each host. Unknown inputs with danger signals are transferred to lymph nodes for further analysis. In larger networks  $l < n \wedge l > 1$  is true, where  $l$  is the set of hosts with lymph nodes and  $n$  is the quantity of hosts. Host 2 illustrates a single CARDINAL host as proposed by Kim et al. (2005).

## Problem

A problem for moderately sized real-time distributed applications is maintaining enough computational capacity for decision analysis. We are addressing the performance requirement for real-time security applications in industrial control system (ICS) network and supervisory control and

data acquisition (SCADA) network scenarios. These networks commonly have combinations of legacy and new systems that need real-time protection. SCADA networks, by definition are remotely and sparsely located. Balancing computational effort is often a solution to the computation capacity problem. These same applications and networked computer systems are also targeted by hackers and malware. In this case the load balancing solution needs to be robust (no single point of failure) and adaptive to a number of compromised systems or disconnected network segments. Therefore, we need an architecture (like CARDINAL-E) that would support these application scenarios whilst requiring minimal computational resources.

## Approach

To achieve these aims we worked toward a decentralised and fault tolerant framework and selected an immune system inspired approach. The CARDINAL framework against malicious worms by Kim et al. (2005) provided an artificial immune system (AIS) base for this self-organisation and the malware self-healing functionality. CARDINAL offered malleability to network-based decentralised decision making applications and embodied immunological danger theory, see Matzinger (1994). Our work employed the conceptual AIS framework in Stepney et al. (2005) to aid the complex (distributed) systems framework development. This led us to consider the decentralised self-management and self-organisation mappings to immune system flow and signalling regulatory mechanisms and other biological processes including cell potency and conscious awareness. We also considered these mappings to the engineered robustness of recent peer-to-peer botnet architectures particularly for their approaches to minimising their impact on network traffic, see Wang et al. (2010).

## Method

In the proposed method, the key modifications to Kim et al.’s framework are the conditional removal of lymph node behaviour from *some* computers and periodic reorganisation of the behavioural roles. Figure 1 simplifies the proposed architecture. An engineered implementation of CARDINAL-

E will only accept the networked application with multiple Lymph Node components.

The reorganisation process proposed enables the reactive lightweight CARDINAL component (input sensor, signatures, etc.) to execute on all CARDINAL-E hosts and the heavier-weight analysis (diagnosis, etc.) to execute on fewer selected hosts. A reorganisation procedure of distributed system roles requires routines for initialisation, transfer of roles, periodic data analysis, historical data analysis, decentralised decision making, secure data transfer, etc. We expect suitable metrics for the role transfer decision to be CPU, memory, disk space and network load parameters recorded and averaged over given time periods. Further careful consideration of these routines will potentially lead to the occurrence of adaptive homeostasis and the emergence of regulatory network activity cycles for a given network.

### Future Application

A future application of CARDINAL-E is to self-healing network security on industrial network types. This requires definition of the CARDINAL application. Kim et al.'s CARDINAL framework uses a population based model of dendritic cell and T-cell interactions to decide when to respond. The definitions of danger theory damage proposed by Kim et al. define bad or unwanted behaviour. The degree (severity and certainty) to which that unwanted behaviour has been experienced will determine the response category. CARDINAL leaves the definition of *strong* and *weak* responses, and response types open to the application designer. To close some of CARDINAL's open representations, Fu et al. (2007) introduced the MAAIS architecture as a derivative of CARDINAL. We intend to reuse Fu et al.'s following additions "length of attack time" to CARDINAL's damage certainty and severity factors; a two-tier damage threshold to decide whether attack analysis requests are handled locally (client) or remotely (server); a weighted equation to determine a damage value; and use of the ASK and TELL protocol for agent communications.

Furthermore, we describe the key representations of detection (antigen input vector), diagnosis (innate system damage decision vector) and experimental response. We represent antigen input as a statistically dimensionally reduced set of eight attributes from the KDD Cup '99<sup>1</sup> dataset. These attributes are {*protocol\_type*, *service* (*port number*), *flags*, *src\_bytes* (*received*), *dst\_bytes* (*sent*), *count*, *diff\_srv\_rate*, *dst\_host\_same\_src\_port\_rate*} which are collected over two second intervals. Appropriate attributes are summed over the interval using a connection identifier defined as  $\langle src\_ip+port+dst\_ip+port \rangle$ . We represent Kim et al.'s damage (necrosis) signal as network packet latency (to

local area and Internet hosts) and individual process loads (CPU and memory loads) where their values breach given thresholds. At this experimental stage, we use Kim et al.'s three effector T-cells to label responses. These are strong response (CD8<sup>+</sup> or cytotoxic T-cells), weak response (helper 2 T-cells) and assist a strong response (helper 1 T-cells), Kim et al. (2005).

### Conclusion

This work leads toward a self-healing system against novel attacks and malware on ICS and SCADA network scenarios, such as dynamic military and defence computer networks and manufacturing industrial networks of various scales. Further work is required to define the CARDINAL-E reorganisation routines and the immune system inspired decentralised decision making, and the CARDINAL diagnosis and response mechanism for novel attacks must be explicitly addressed.

### Acknowledgements

This work was supported by EPSRC I-CASE studentship EP/J501785/1 and EADS Innovation Works Program IW201339.

### References

- Fu, H., Yuan, X., and Wang, N. (2007). Multi-agents artificial immune system (maais) inspired by danger theory for anomaly detection. In *Computational Intelligence and Security Workshops, 2007. CISW 2007. International Conference on*, pages 570–573.
- Kim, J., Wilson, W., Aickelin, U., and McLeod, J. (2005). Cooperative automated worm response and detection immune algorithm (cardinal) inspired by t-cell immunity and tolerance. In *Artificial Immune Systems. ICARIS 2005. International Conference on*, pages 168–181. Springer.
- Matzinger, P. (1994). Tolerance, danger, and the extended family. *Annu Rev Immunol*, 12:991–1045.
- Stepney, S., Smith, R. E., Timmis, J., Tyrrell, A. M., Neal, M. J., and Hone, A. N. W. (2005). Conceptual frameworks for artificial immune systems. *International Journal of Unconventional Computing*, 1(3):315–338.
- Wang, P., Sparks, S., and Zou, C. C. (2010). An advanced hybrid peer-to-peer botnet. *IEEE Transactions on Dependable and Secure Computing*, 7(2):113–127.

<sup>1</sup>Computer Network Intrusion Detection DARPA Competition Dataset 1999, University of California, Irvine. <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html> (Accessed: 27/02/2013)

- Adami, Christoph, 126, 1067  
Aguilera, Miguel, 51, 395  
Aickelin, Uwe, 854, 891  
Alderton, Simon, 27  
Alers, Sjriek, 775  
Amaducci, Matteo, 783  
Amos, Martyn, 925  
Andrew, Angeline, 94  
Anthony, Neil, 829  
Antonioni, Alberto, 226  
Arena, Paolo, 765  
Arita, Takaya, 425, 439, 617  
Arnold, David, 79  
Arnold, Solvi F., 425  
Arriaga, Julio G., 827  
Asada, Minoru, 324  
Attwood, Sue, 633  
Aubert, Nathanael, 1144  
Augusto, Douglas, 854  
Austin, James, 71  
Azumagakito, Tsubasa, 617  
Baccouche, Alexandre, 791  
Baddeley, Bart, 1007  
Bajec, Iztok Lebar, 1114  
Bakkum, Douglas, 1075  
Balch, Tucker, 686  
Ball, Gordon, 1223  
Banzhaf, Wolfgang, 981  
Baptista, Tiago, 554  
Barandiaran, Xabier E., 395  
Barbosa, Helio, 854  
Bate, Iain, 846  
Batut, Bérénice, 43  
Baugh, Declan, 210  
Bazzan, Ana L. C., 909  
Bedia, Manuel G., 395  
Beer, Randall D., 51, 268  
Benedettini, Stefano, 372  
Beslon, Guillaume, 43  
Biehl, Martin, 1099  
Birattari, Mauro, 783  
Bourreau, Eric, 997  
Boursin, Francois Asperti, 1009  
Brede, Markus, 1, 159, 300, 472, 593  
Bredeche, Nicolas, 260, 671, 1056, 1144  
Brom, Tim, 126  
Bruger, Eric, 126  
Buesser, Pierre, 192  
Bukovac, Zoe, 276  
Bullock, Seth, 151  
Cañamero, Lola, 505, 633  
Cabri, Giacomo, 864  
Calcraft, Paul, 1107  
Campo, Alexandre, 102  
Cangelosi, Angelo, 773  
Cannon, Judy, 1009  
Capodici, Nicola, 864  
Carrara, Paolo, 1161  
Cavalcante Neto, Joaquim Bento, 577, 750  
Cazalis, Roland, 1233  
Cejkova, Jitka, 818  
Channon, Alastair, 973  
Chapman, Samuel, 1067  
Chavez, Vasthi Alonso, 625  
Chen, Beibei, 876  
Chiesa, Stefano, 1049  
Clarke, Sebastian, 1017  
Cleaves, Henderson, 829  
Cody, Martin L., 827  
Coghill, George, 957  
Collard, Philippe, 480  
Correia, Luis, 1116  
Costa, Ariadne, 940  
Costa, Ernesto, 554  
Cotta, Carlos, 523  
Crailsheim, Karl, 609, 721, 1041  
Cronin, Leroy, 1066, 1132, 1142  
Crossley, Matthew, 925  
D'Aguanno, Erica, 1161  
Damiani, Chiara, 793  
Darabos, Christian, 340, 348  
de Brito, Carlos Eduardo Fisch, 577, 750  
de Peer, Yves Van, 1235  
De Pinedo, Manuel, 665  
Demšar, Jure, 1114  
Der, Ralf, 601  
Di Mario, Ezequiel, 561  
Di Nuovo, Alessandro, 773  
Di Nuovo, Santo, 773  
Dinh, Quang Huy, 1144  
Disso, Jules Pagna, 1236  
Dittrich, Peter, 248  
Doncaster, C. Patrick, 167, 625  
Dorigo, Marco, 102  
Dorin, Alan, 276  
Doursat, Rene, 454  
Ducharme, Vincent, 802  
Dyer, Adrian, 276  
Dyke, James G., 387, 625  
Egbert, Matthew, 248  
Egli, Richard, 802  
Egloff, Mattia, 226  
Eiben, A.E., 1056  
Ellefsen, Kai Olav, 292, 409  
Erdei, Joachim, 531  
Erskine, Adam, 35  
Escuela, Gabi, 248  
Fahr, Alfred, 1161  
Falahat, Darius, 593  
Fellermann, Harold, 1083  
Ferber, Amber, 521  
Fernandes, Carlos, 523  
Ferrauto, Tomassino, 538  
Ferreira, António Homem, 649  
Figueredo, Graziela, 854  
Filisetti, Alessandro, 372, 793  
Fioriti, Vincenzo, 1049  
Flamm, Christoph, 1152  
Fontana, Alessandro, 144  
Francesca, Gianpiero, 379  
Fraticini, Fabio, 1049  
Frenoy, Antoine, 200  
Frey, Urs, 1075  
Fricke, Matthew G., 1009  
Fujii, Naotaka, 810  
Fujii, Teruo, 791, 1144  
Fujiwara, Kei, 208, 488  
Gürçan, Önder, 1173  
Gagne, Caroline, 933  
Ganascia, Jean Gabriel, 1207  
Gao, Shangce, 876  
Garattoni, Lorenzo, 783  
Georgiou, Loukas, 917  
Gerding, Enrico, 593  
Gerrard, Claire E., 957  
Giacobini, Mario, 340  
Gigliotta, Onofrio, 538  
Glackin, Cornelius, 118  
Goings, Sherri, 1091  
Goles, Eric, 1085  
Gong, Tao, 876, 901  
Gonzalez, Miguel, 167  
Gordon Smith, Chris, 1225  
Gorny, Damian, 818  
Graham, Britney E., 348  
Graham, Paul, 1007  
Graudenzi, Alex, 793  
Gravel, Marc, 933  
Greensmith, Julie, 891  
Groote, Jan Friso, 232  
Gruenert, Gerd, 248  
Guarneri, Andrea, 585  
Gunji, Yukio, 492, 810  
Gutmann, Anne, 11  
Haasdijk, Evert, 671  
Hagiya, Masami, 1144  
Haidar, Alaa Abi, 1207  
Hamann, Heiko, 947  
Harada, Tomohiro, 713  
Hart, Emma, 856, 864, 891  
Haruna, Taichi, 175  
Harvey, Inman, 1171  
Hasegawa, Tomonori, 316  
Haufová, Petra, 818, 1064  
Hecker, Joshua, 1009, 1025  
Hemmingsen, Chris, 1152  
Heras Escribano, Manuel, 665  
Herrmann, J. Michael, 35, 641  
Hervouet, Fabien, 997  
Hessel, Jack, 1091  
Hickinbotham, Simon, 71  
Hintze, Arend, 126, 1067  
Honda, Manabu, 250  
Hoogendoorn, Mark, 1056  
Hrbacek, Radek, 431  
Hrolenok, Brian, 686  
Hu, Ting, 94, 348  
Hubert, Julien, 698, 1075  
Humphreys, Ben, 965  
Husbands, Phil, 513, 1007, 1107  
Ianni, Antonella, 472  
Iba, Hitoshi, 1144  
Ichinose, Genki, 1221  
Ikegami, Takashi, 240, 498, 698, 1075  
Imai, Michita, 692  
Impett, Jonathan, 1126  
Impett, Leonardo, 1126  
Inden, Benjamin, 184  
Infantino, Ignazio, 1005, 1192  
Ito, Takashi, 439  
Izquierdo, Eduardo J., 51  
Jackson, Adam, 356  
Jetté, Sylvie, 802  
Joachimczak, Michal, 531, 454  
Jost, Jürgen, 184  
Jovanovic, Kosta, 136  
Kaminker, Vitaliy, 825  
Karagas, Margaret, 94  
Kato, Kazuhiko, 498  
Kato, Takeo, 758  
Kauranne, Tuomo, 657  
Kawai, Norie, 250  
Keat, William, 965  
Keijzers, Stein, 838  
Kengyel, Daniela, 1041, 1124  
Keshishzadeh, Sarmen, 232  
Khaluf, Yara, 737  
Khazanov, Mark, 965  
Kiryzov, Kiril, 19  
Knibbe, Carole, 43  
Knoester, David, 126, 1067  
Komatsu, Daiki, 208  
Kossan, George, 827  
Kostic, Dragan, 136  
Kowaliw, Taras, 454  
Kuruma, Yutetsu, 9, 824  
Kuyper, Ida Sprinkhuizen, 838  
Labrosse, Frederic, 1017  
Lancet, Doron, 134  
Lane, David Avra, 372  
Laredo, Juan, 523  
Lau, Henry Y. K., 872, 899  
Lau, Huikeng, 846  
Lau, Wai Fai, 521  
Lee, Nicole, 899  
Leonova, Hannah, 874  
Letendre, Kenneth, 1025  
Li, Cai, 1183  
Lones, John, 505  
Lowe, Robert, 19, 1183  
Luisi, Pier Luigi, 1161  
Luo, Wenjian, 884  
Maandag, Peter, 838  
Mackenzi, Craig O., 340  
Macleod, Christopher, 957  
Maekawa, Tadao, 250  
Maggiorini, Dario, 585, 706  
Maher, John, 446  
Manoonpong, Poramate, 989  
Marchal, Kathleen, 1235  
Marchiori, Elena, 838  
Mariano, Pedro, 1116  
Markovitch, Omer, 134  
Marocco, Davide, 679, 773  
Martinho, Carlos, 649  
Martinoli, Alcherio, 561

- Martius, Georg, 601  
 Maselko, Jerzy, 825  
 Massera, Gianluca, 538  
 Matsubayashi, Hideaki, 9, 824  
 Matsuda, Eiko, 698, 1075  
 Matsuoka, Yoshiyuki, 758  
 Matyas, Vashek, 569  
 May, Stefan, 775  
 Mccall, John, 957  
 McEwan, Chris, 891  
 McGowan, Craig, 11  
 Mckay, Bob, 332  
 McKinley, Philip, 11  
 McMullin, Barry, 210, 316  
 Meister, Tobias, 1124  
 Merelo, Juan Julian, 523  
 Merkle, Daniel, 1152  
 Michiels, Nico, 1033  
 Miletitch, Roman, 102  
 Mirazo, Kepa Ruiz, 1163  
 Mirmomeni, Masoud, 126  
 Misevic, Dusan, 200  
 Mita, Takeshi, 1075  
 Mitavskiy, Boris, 379  
 Miyashita, Shuhei, 955  
 Montalva, Marco, 1085  
 Montanier, Jean Marc, 260  
 Moore, Jared, 11  
 Moore, Jason H., 59, 94, 110, 340, 348, 364  
 Moran, Timothy, 472  
 Morer, Ignacio, 395  
 Mori, Hiroki, 324  
 Moses, Melanie, 1009, 1025  
 Murakami, Hisashi, 492  
 Murnik, Maja, 258  
 Nakamura, Tomoaki, 810  
 Navarro, Iñaki, 561  
 Neal, Mark, 1236  
 Nichele, Stefano, 63  
 Niizato, Takayuki, 729  
 Nisbet, Andy, 925  
 Nishina, Emi, 250  
 Nitschke, Geoff, 1219  
 Noble, Jason, 27, 159, 167, 472, 665  
 Nogueira, Yuri Lenon Barbosa, 577, 750  
 Nolfi, Stefano, 538  
 Nomura, Shin Ichiro M., 208, 488  
 Oka, Mizuki, 498, 1075  
 Okuyama, Yuzi, 324  
 Olah, Mark, 1134  
 Olson, Randal, 126  
 Oohashi, Tsutomu, 250  
 Osawa, Hirotaka, 692  
 Pacheco, Jorge M., 61, 182  
 Padirac, Adrien, 791  
 Paechter, Ben, 856  
 Pan, Qinxin, 94  
 Pantaleone, James, 825  
 Panzeri, Samuele, 706  
 Parsons, David P., 43  
 Patanè, Luca, 765  
 Payne, Joshua, 59, 364  
 Pechenick, Dov, 59, 364  
 Petric, Špela, 258  
 Petrovic, Vladimir M., 136  
 Philippides, Andrew, 1007  
 Pilat, Marcin, 439  
 Pilato, Giovanni, 1005  
 Pinciroli, Carlo, 783  
 Pinheiro, Flavio L., 61, 182  
 Pitonakova, Lenka, 151  
 Pliszka, Zbigniew, 663  
 Polack, Fiona, 893  
 Polani, Daniel, 118, 1099  
 Potkonjak, Veljko, 136  
 Pugliese, Francesco, 679  
 Punithan, Dharani, 332  
 Radenski, Atanas, 284  
 Rammig, Franz, 737  
 Rasmussen, Steen, 521, 1083  
 Read, Mark, 874, 891  
 Ribeiro, Andre S., 1175  
 Rieffel, John, 965  
 Ripamonti, Laura Anna, 585, 706  
 Rizzo, Riccardo, 1005, 1192  
 Roli, Andrea, 372, 783  
 Rondelez, Yannick, 791, 1144  
 Rosa, Agostinho, 523  
 Rose, Louis, 874  
 Rosenkind, Micah, 79  
 Ross, Philipp, 521  
 Rus, Daniela, 955  
 Ruz, Gonzalo, 1085  
 Saberi, Arash Khabbaz, 232  
 Sahraei, Bijan Ranjbar, 462, 775  
 Saito, Ken, 546  
 Saito, Masaya, 1221  
 Sakiyama, Tomoko, 810  
 Saleh, Adam, 569  
 Salge, Christoph, 118  
 Santos, Francisco C., 61, 182  
 Sasaki, Takahiro, 470  
 Sato, Koichiro, 488  
 Sato, Yusuke, 758  
 Sayama, Hiroki, 521, 1221  
 Schmickl, Thomas, 609, 721, 830, 1041, 1124  
 Schneider, Dominique, 43  
 Schwarzer, Christopher, 1033  
 Scully, Peter, 1236  
 Segerstahl, Margareta, 745  
 Semenov, Oleg, 1134  
 Serra, Roberto, 372, 793  
 Shcherbacheva, Anna, 657  
 Shimizu, Yoshihiro, 1160  
 Shirt Ediss, Ben, 1163  
 Siebers, Peer Olaf, 854  
 Sikulova, Michaela, 431  
 Sim, Kevin, 856  
 Singh, Mandeep, 1064  
 Sirmai, Jean, 417  
 Six, Adrien, 1207  
 Smith, Simón C., 641  
 Snowdon, James, 308  
 Solé, Ricard, 1163  
 Song, Jingping, 1236  
 Spranger, Michael, 1200  
 Stankova, Katerina, 462  
 Stanley, Kenneth, 218  
 Stano, Pasquale, 1161  
 Stanton, Adam, 973  
 Stefanovic, Darko, 1134  
 Stehlík, Martin, 569  
 Štěpánek, František, 818, 1064  
 Stetsko, Andriy, 569  
 Stettiner, Orly, 403  
 Stojanovic, Milan, 1134  
 Stolleis, Karl, 1025  
 Stradner, Juergen, 1056  
 Strauss, Roland, 765  
 Suzuki, Reiji, 425, 439, 617  
 Swenson, Bjorn, 1025  
 Szerlip, Paul, 218  
 Taddei, François, 200  
 Takadama, Keiki, 713  
 Takahama, Shiho, 546  
 Takahashi, Hirokazu, 1075  
 Takato, Minami, 546  
 Tan, Jie, 110  
 Tanaka, Shinpei, 1083  
 Tanaka, Yo, 1160  
 Tanida, Junichi, 546  
 Taylor, Charles E., 827  
 Taylor, Paul, 893  
 Te, Rene, 633  
 Tegner, Jesper, 1223  
 Thenius, Ronald, 830, 1041, 1124  
 Thomas Vaslin, Veronique, 1207  
 Thorniley, James, 513  
 Timmis, Jon, 846, 874, 893  
 Tinós, Renato, 940  
 Tomassini, Marco, 192, 226, 340  
 Torres, Andre Estevez, 791  
 Tran, Huy, 1175  
 Trianni, Vito, 102, 1017  
 Tripp, Magnus, 874  
 Trubian, Marco, 585  
 Tsang, Wilburn W. P., 872  
 Tuci, Elio, 379, 1017  
 Tudge, Simon, 1, 300  
 Tufte, Gunnart, 63  
 Turšič, Miha, 258  
 Tuyls, Karl, 462, 775  
 Tyrrell, Andy, 1056  
 Uchikoba, Fumio, 546  
 Ueda, Takuya, 9, 824  
 Ueno, Osamu, 250  
 Ullrich, Martin, 1064  
 Unold, Olgierd, 663  
 Urbano, Paulo, 917  
 Urbanowicz, Ryan, 110  
 Vallejo, Edgar E., 823  
 Vardy, Andrew, 981  
 Vargas, Patrícia, 940  
 Vasconcelos, Vítor V., 61, 182  
 Vella, Filippo, 1005  
 Vidal, Creto Augusto, 577, 750  
 Villani, Marco, 372, 793  
 Virgo, Nathaniel, 240  
 Vorobyev, Gregory, 981  
 Wakisaka, Sohei, 810  
 Wang, Fang, 876  
 Waterson, Ben, 308  
 Watson, Richard, 1, 167, 356  
 Weaver, Iain, 387  
 Weeks, Michael, 71  
 Weiss, Gerhard, 462, 775  
 Williams, Lance, 86  
 Williams, Paul, 268  
 Wilson, David Sloan, 1221  
 Winfield, Alan, 1056  
 Winstanley, Graham, 79  
 Witkowski, Olaf, 1219  
 Woergoetter, Florentin, 989  
 Woodward, Alex, 498  
 Wrobel, Borys, 144, 454, 531  
 Wu, Binglu, 901  
 Xiong, Xiaofeng, 989  
 Yamasaki, Shinpei, 546  
 Yao, Yao, 1235  
 Yu, Lingli, 901  
 Zahadat, Payam, 609, 721, 830  
 Zenil, Hector, 1223  
 Zhao, Dongdong, 884  
 Zhao, Wyman, 521  
 Zhong, Weiya, 876  
 Zhu, Yiqing, 498  
 Ziemke, Tom, 1183  
 Zinflou, Arnaud, 933